HEALTH ECONOMICS RESEARCH, INC.

300 Fifth Avenue, 6th Floor
Waltham, MA 02154
(617) 487-0200
(617) 487-0202 Fax

UNIQUE PHYSICIAN IDENTIFICATION NUMBER (UPIN) VALIDATION STUDIES

Researcher Edits

Submitted by:

Killard W. Adamache, Ph.D.
William R. Scott, III

Prepared for:

Ann Meadow, Sc.D.
HCFA Project Officer

August 24, 1994

This research was funded under Delivery Order No. 5 of the HCFA Master Contract No. 500-92-0020. The statements contained in this report are solely those of the authors and do not necessarily reflect the views or policies of HCFA. The contractor assumes responsibility for the accuracy and completeness of information contained in this report.

UNIQUE PHYSICIAN IDENTIFICATION NUMBER (UPIN) VALIDATION STUDIES

PREFACE

The *Physician Registry* was established in 1989 by HCFA for the purpose of issuing unique identification numbers (UPINs) to the following Medicare providers: doctors of medicine, osteopathy, dental medicine, dental surgery, podiatry, and optometry and chiropractors. To assist in the identification of individual providers, the *Registry* collects a variety of information. HCFA contracted with Health Economics Research, Inc. (HER) to perform studies on the *Registry's* data quality and on the consistency of physician information on Medicare claims. Another objective was to develop data edits to assist carriers in maintaining their *Registry* data and to assist researchers in using *Registry* data.

The current report is the third produced under the HER contract. It provides data edits for use by researchers using *Registry* data. The first report describes the results of an analysis of the quality of *Registry* data. The second report provides data edits for use by carriers in maintaining their *Registry* data. The fourth and final report has several components, including: (1) case studies of the operations of the *Registry* contractor and carrier operations, (2) an analysis to identify the inadvertent issuance of UPINs to group practices and corporations, (3) estimates of the number of HMO physicians without UPINs who provide services to Medicare beneficiaries, and (4) an analysis of the consistency of a physician's Medicare participation status on the *Registry* and on the physician's associated Medicare claims. An additional deliverable consists of documentation for SAS files submitted to HCFA. The SAS files contain special variables that indicate the quality of data elements maintained by the *Registry*, the consistency between pairs of data elements, and whether a provider might have more than one UPIN.

TABLE OF CONTENTS

PAGE

1.0 INTRODUCTION 1
- 1.1 Overview of Registry Data Elements 1
- 1.2 File and Data Formats 2
- 1.3 Overview of Researcher Edits 3
- 1.4 Use of the Programming Specifications 6

2.0 INDIVIDUAL VARIABLE EDITS 8
- 2.1 Introduction 8
- 2.2 Logic to Create Base Record 8
- 2.3 Individual Business Setting Records 10
- 2.4 Base Record Variables 11
  - 2.4.1 Unedited Variables 11
  - 2.4.2 Edited Variables 11
- 2.5 Business Setting Variables 13
  - 2.5.1 Unedited Variables 14
  - 2.5.2 Edited Variables 14

BASE RECORD: UNEDITED DATA ELEMENTS 15

Data Element: Sanction Code 16
Data Element: Sanction Date 19
Data Element: Sanction Length 22

BASE RECORD: EDITED DATA ELEMENTS 25

Data Element: Birth Year 26
Data Element: Date of Birth 29
Data Element: Death Year 33
Data Element: Date of Death 36
Data Element: School Code 40
Data Element: Graduation Year 54
Data Element: Provider Credentials 57
Data Element: Physician Status 60

BUSINESS SETTING RECORD: EDITED DATA ELEMENTS 63

Data Element: State License Number 64
Data Element: State Licensed In 67
Data Element: Specialty Code 70
Data Element: Board Certification 73
Data Element: DRIP Status 76
Data Element: Group Practice Indicator 79
Data Element: Physician Participation Indicator 82

TABLE OF CONTENTS (continued)

PAGE

3.0 EDITS TO OBTAIN LOGICAL CONSISTENCY BETWEEN VARIABLES.....85

4.0 DETECTION OF MULTIPLE-UPIN PROVIDERS 118

4.1 Multiple-UPIN Providers 118

4.2 Screening Variable Data Elements 119

4.3 Results of Implementing the Four Detection Algorithms 120

4.4 Provider Records Without Feasible Dates of Birth..... 126

TABLE OF EXHIBITS

		PAGE
EXHIBIT 1-1	FORMATS OF REGISTRY DATA ELEMENTS	4
EXHIBIT 2-1	REGISTRY SCHOOL CODES	41
EXHIBIT 3-1	SCHOOL RECODES	88
EXHIBIT 3-2	LOGICALLY CONSISTENT VALUES OF PROVIDER CREDENTIALS AND SCHOOL RECODE (MSCHTYPE)	89
EXHIBIT 3-3	CREDENTIAL-BASED RECODES FOR SCHOOL CODE	90
EXHIBIT 3-4	RECODE OF SPECIALTY CODE	105
EXHIBIT 3-5	LOGICALLY CONSISTENT VALUES OF PROVIDER CREDENTIALS (ECRED) AND SPECS	106
EXHIBIT 3-6	ASSIGNMENT OF OSTEOPATHIC SPECIALTY CODE FOR ESPEC1B	107
EXHIBIT 3-7	CREDENTIAL-BASED RECODES FOR SPECIALTY CODE	108
EXHIBIT 3-8	LOGICALLY CONSISTENT VALUES OF SPECIALTY RECODE (SPECS) AND SCHOOL RECODE (MSCHTYPE)	115

TABLE OF APPENDICES

		PAGE
APPENDIX A	SAS PROGRAM TO CREATE BASE RECORDS	128
APPENDIX B	SAS PROGRAM TO EDIT BUSINESS SETTING VARIABLES	138
APPENDIX C	SAS PROGRAM FOR LOGICAL CONSISTENCY EDITS	141
APPENDIX D	SAS PROGRAMS FOR DETECTING MULTIPLE-UPIN PROVIDERS	151
APPENDIX D-1	ID_KEY_1 = DOB, L1_3, F1, LNAMESUM	152
APPENDIX D-2	ID_KEY_2 = DOB, L1_3, LNAMESUM	158
APPENDIX D-3	ID_KEY_3 = DOB, L1_3, F1	164
APPENDIX D-4	ID_KEY_4 = DOB, L1_3	169
APPENDIX D-5	ID_KEY_5 = L1_3	174
APPENDIX E	MERGING PROGRAMS	179
APPENDIX E-1	PROGRAM TO MERGE MULTIPLE UPIN INDICATORS WITH MAIN ANALYTIC FILE	180
APPENDIX E-2	PROGRAM TO MERGE EDITED FILE WITH ORIGINAL FILE	182
APPENDIX F	MEDICAL SCHOOL ESTABLISHMENT AND CLOSURE DATES	184

1.0 INTRODUCTION

The *Physician Registry* was established in 1989 by HCFA for the purpose of issuing unique identification numbers (UPINs) to the following Medicare providers –doctors of medicine, osteopathy, dental medicine, dental surgery, podiatry, and optometry and chiropractics. To assist in the identification of individual providers, the *Registry* collects a variety of information.

The *Registry's* data has been analyzed in our *Carrier Analysis.*[1] In performing the analysis, algorithms were developed to detect Missing, Erroneous, Dubious, and Unknown (MEDU) values of data elements. The current report was developed to provide researchers with a set of algorithms to edit *Registry* data elements. Also provided are algorithms to detect and resolve logical inconsistencies between pairs of data elements (e.g., is the specialty code consistent with the provider's credentials?). Finally, algorithms are provided to detect providers with more than one UPIN.

The next section provides a brief overview of *Registry* data elements. Section 1.2 discusses file and data formats. Section 1.3 provides a brief overview of the edits and programs presented in this report. Section 1.4 provides guidance on how to interpret the edit specifications, flowcharts, and computer programs presented in Chapters 2 through 4.

1.1 Overview of *Registry* Data Elements

To maintain the *Registry*, Medicare carriers forward provider information to the *Registry* contractor, Transamerica Occidental Life Insurance Company (TOLIC). There are two basic types of provider information maintained by the Registry: (1) information to help identify

[1]Unique Physician Identification Number (UPIN) Validation Studies: Carrier Analysis, HCFA Master Contract No. 500-92-0020, Delivery Order No. 5, August 23, 1994.

providers so that only one UPIN is issued to a provider and (2) payment and participation information (e.g., specialty, Medicare participation status, and sanction status). Among the data elements collected by the *Registry* are name, date of birth, date of death (if deceased), medical school from which the provider's degree was granted,[2] year graduated from medical school, sanction information, credentials (e.g., MD or DDS), specialty code, Medicare participation status, type of practice (i.e., solo or group), and board certification.

1.2 File and Data Formats

The data used in the *Carrier Analysis* were sent to HCFA by TOLIC. TOLIC sends an updated version of the Registry data to HCFA on a quarterly basis. The July 1993 version was used in the *Carrier Analysis*; and it was used for testing the programs included in this report.

As received by HCFA, the file consists of variable-length records. There is one record per provider on the file. Each record has a *header* section followed by a *practice setting* section that contains data on one or more practice settings for that provider. Since the number of practice settings varies by provider, the length of the *practice setting* section varies in direct proportion to the number of practice settings.

For our developmental work, we used a HCFA version of the Registry data in which the variable-length records had been converted to fixed-length records. Each fixed-length record consists of two major sections: (1) data from the *header* section of TOLIC's file and (2) data from *one* of the practice settings. In the file with fixed-length records, there are as many records for a given provider as the provider has practice settings.

There is some overlap in the data elements contained in the header and practice setting sections of each record. Provider name information, sanction information, dates of birth and

[2] In this context, *medical* school refers to a school which grants one of the following degrees: Doctor of Medicine, Doctor of Osteopathy, Doctor of Dental Surgery, Doctor of Dental Medicine, Doctor of Podiatry, Doctor of Optometry, and chiropractors. In Chapter 3, *medical* school only refers to schools which grant the Doctor of Medicine degree.

death, medical school, and year of graduation are present in both sections. The values of common data elements may differ in different locations (e.g., header versus setting).

The editing algorithms were designed to meet HCFA's specifications for records that carriers send to the *Registry* ("Carrier Record" in Exhibit 1-1). Some of the data elements in the HCFA file have different COBOL pictures; for example, the date of birth components have different order ("HCFA File" in Exhibit 1-1). As a result, our SAS programs had to account for data contingencies that may not be present in certain carrier provider files (e.g., secondary specialty code an otherwise numeric field that might contain all blanks) but are present in the file TOLIC provides to HCFA.

1.3 Overview of Researcher Edits

Three types of edits are indicated in this report. Edits to individual variables are described in Chapter 2. Most of these edits do not take into consideration values of other variables.[3] There are two types of individual variable edits. The first is for variables that should have the same value regardless of the provider's practice setting (e.g., birth date). The second is for variables whose values may vary by practice setting (e.g., Medicare participation status). Edits to detect and resolve logical inconsistencies between pairs of variables are described in Chapter 3. An example of a logical inconsistency is when the school code indicates that the provider attended a chiropractic school while the credentials indicate that the provider is a dentist. Chapter 4 describes algorithms to detect providers that have more than one UPIN. A discussion of the strengths and weaknesses of the algorithms, based on our *Carrier Analysis*, is also provided.

In addition to the edit programs, we also provide programs to create a new analytic file for researchers. The new analytic file has a structure identical to the original HCFA file. In

[3]The exception is for birth year and graduation year.

EXHIBIT 1-1

FORMATS OF REGISTRY DATA ELEMENTS

	CARRIER RECORD					*HCFA FILE*	
Data Element	Required	COBOL Picture	Justification	Component Order	Missing Value	COBOL Picture	Component Order
Date of birth	Y	9(06)	R	MMDDYY	0	S9(09) COMP-3	YYMMDD
Date of death	Y	9(06)	R	MMDDYY	0	S9(09) COMP-3	YYMMDD
School code	Y	X(05)	R		blank	X(05)	
Year graduated	Y	9(02)	R	YY	0	S9(09) COMP-3	YY
Sanction code	N	X(01)			blank	X(01)	
Sanction, effective date	N	9(04)	R	MMYY	0	9(04)	MMYY
Sanction, number of years	N	9(02)	R		0	S9(03) COMP-3	
State license number	Y	X(12)	R		blank	X(12)	
Physician status	Y	9			0	X(01)	
Provider credentials	Y	X(03)	L		blank	X(03)	
Primary specialty	Y	9(02)	R		0	X(02)	
Secondary specialty	N	9(02)	R		0	X(02)	

Sources:

1. Carrier record formats from HCFA Transmittals.
2. Registry file formats from Transamerica Occidental Life Insurance Company: version of file sent to HCFA.

place of the HCFA file's *header* segment, we create a *base* segment which contains edited variables whose values should not vary by practice setting. In place of the HCFA file's *practice* setting segment, we create a *business* setting segment which contains edited variables whose values might vary by practice setting. (To avoid confusion in the discussion about the original variables in the HCFA file or the edited variables in the new analytic file, the terms *base* and *business* are used in place of, respectively, *header* and *practice* for the record segments in the new analytic file.)

Before implementing the computer programs described herein, it is advisable to first create a SAS file from the 515-byte flat file created by HCFA's Bureau of Data Management and Strategy (BDMS).[4] The program that we used to create a SAS file is available from HCFA. The names assigned to the SAS variables were based on the TOLIC data element names.

There are several steps involved in creating the new analytic file. A base record is first created (Chapter 2). Next, individual business setting records are created (Chapter 2). The base records and business setting records are merged (by UPIN) to create an analytic file that has the same structure as HCFA's file (Chapter 3). Because providers have multiple practice settings, there are three to four times as many business records as there are base records. Therefore, simple one-to-one merges are not possible.

The program in Appendix C has the appropriate code to properly merge the base and business records. The multiple-UPIN indicators developed in Chapter 4 are consolidated in one file (the last part of the program in Appendix D-5) and that file can then be merged to the analytic file using the program in Appendix E-1. Finally, if the researcher wishes to merge the analytic file to the original, unedited file, the program in Appendix E-2 can be used. To properly merge the two files, a special identifier to uniquely identify each business record is created.

---

[4]For our developmental work, we used the following flat file created by BDMS: DSN=MU00.@AAA2365.XUPIN.JUL0293.FIX. The HCFA DSN of the SAS program that creates the SAS file is: RR29.@DCH7910.HER.UPIN.LIB(APR40494).

1.4 Use of the Programming Specifications

Data integrity specifications, flowcharts, and SAS computer programs are provided to assist programmers in implementing the edit algorithms. In the flowcharts and SAS programs, *flag* variables are used to indicate data element integrity.

Our SAS programs had to account for data possibilities that might not be present in a carrier's provider file (e.g., secondary specialty code, an otherwise numeric field that might contain all blanks). Except for such circumstances, the flowcharts should be directly usable by programmers. The STOP statements in the flowcharts indicate that no additional checking of the data element is necessary and that the program can flow to the next step (e.g., check another data element).

Researchers that have SAS can use our programs with few, if any, modifications. Although some researchers may not have SAS, there are many similarities between SAS's IF/THEN/ELSE statements and those found in other programming languages such as COBOL.

When reading a flat (raw) file, if SAS encounters a blank field (regardless of whether the field is numeric or alphanumeric), SAS automatically assigns a *missing value* code to the data element. For numeric fields the missing value code is equal to a period while for alphanumeric fields the missing value code is a blank. If a data element was filled with all 9s instead of zeroes or blanks, it was also considered to be a missing value (except for date of death). In the provided SAS print-outs, a "^" denotes a logical NOT operation, a "|" denotes a logical OR operation and a "||" denotes a concatenation operation.

We have attempted to adhere to standard flowcharting rules. From decision "diamonds," all "*Yes*" branches flow downward and all "*No*" branches flow to the right. However, because of the availability of more recent programming tools and changes in programming philosophy, flowcharts don't always move downward. With regard to IF statements, in particular, conditions that most of the observations satisfy are usually placed at

the beginning of a series of IF/THEN/ELSE statements according to current programming philosophy.

In contrast with traditional, structured computer languages such as COBOL and FORTRAN, SAS is not a structured language. Although there are many similarities between SAS and computer languages such as COBOL, SAS statements can be quite different from statements in other computer languages (e.g., data subsetting statements) and allow operations (e.g., merging two files) that would have require substantially more program code in other computer languages (especially COBOL prior to 1980). Thus, the provided SAS code achieves the objective specified in the flowcharts, but not necessarily in the same order. For example, see the flowchart and SAS code for the integrity of school codes.

2.0 INDIVIDUAL VARIABLE EDITS

2.1 Introduction

The edits for two types of variables are presented in this chapter (Table 2-1): (1) base record (header) variables and (2) business (practice) setting variables. The base record variables are those variables whose values should not vary by business setting. Business setting variables are those variables whose values can vary by business setting.

The following section describes the logic used to create base records. Section 2.3 describes the logic used to create business setting records. Section 2.4 discusses base record variables, and section 2.5 discusses business setting variables. Finally in section 2.5 we provide: (1) algorithms to classify values of the base record sanction variables, (2) edits for other base record variables, and (3) edits for business setting variables.

2.2 Logic to Create Base Record

The objective of the base record variable edits is to have, for each variable, a value that does not vary by business setting. This requires, for each variable, selecting a feasible value which is denoted as the *gold* standard. A *feasible* value is simply a value that satisfies specifications described later in this chapter. The rule employed to find the *gold* standard is as follows. Use the header value (if a header counterpart exists) if it is feasible, otherwise, use the value from the most recent feasible practice setting value. Feasible Railroad Retirement Board (RRB) values are only used if none are available from other carriers.

The logic for the program that creates base records is as follows (see Appendix A for a full program listing). First, observations without a valid date for practice setting addition to the Registry (SETDATE) are deleted. A zero/one binary variable (RRB) is created where zero denotes non-RRB carriers and one denotes RRB carriers. The Registry file is then sorted by

TABLE 2-1

BASE AND BUSINESS SETTING VARIABLES

VARIABLE	DESCRIPTION	EDITED
Base Record Variables		
UPIN	Unique Physician Identification Number	No
ELNAME	Last name	No
EFNAME	First name	No
EMNAME	Middle name	No
ESUFFIX	Name suffix	No
ESANC1	Sanction code	No
ESANCDAT	Sanction effective date	No
ESANCYRS	Years sanctioned	No
EDOBYR	Birth year	Yes
EDOB	Date of birth	Yes
EDODYR	Death year	Yes
EDOD	Date of death	Yes
EMEDSCHL	School code	Yes
EGRADYR	Year of graduation	Yes
ECRED	Provider's Credentials	Yes
EPHYSTAT	Physician's status	Yes
GRADAGE	Provider's age at graduation	Yes
Business Setting Variables		
UPIN	Unique Physician Identification Number	No
CARRIER	Carrier number	No
CPIN	Medicare billing number	No
SETDATE	Date setting added to Registry	No
BILLZIP	Zip code-billing	No
BIZZIP	Zip code-business	No
ELICENSE	State license number	Yes
ESTLICNS	State in which physician is licensed	Yes
ESPEC1	Primary specialty code	Yes
ESPEC2	Secondary specialty code	Yes
EBOARD1	Primary board certification	Yes
EBOARD2	Secondary board certification indicator	Yes
DRIP	Resident or intern code	Yes
EPRCTIND	Group practice code	Yes
EPRT1IND	Physician's participation indicator	Yes

RRB (ascending order) and SETDATE (descending order).

For the first record for each provider, feasible header values of variables are used. If a feasible header value is not found, then the practice setting value is used if it is feasible. If there is more than one practice setting, then subsequent values are checked if a feasible value had not been previously found. If, after checking all of a provider's practice settings, feasible values for birth year or graduation year have not been found, then, if possible, birth year or graduation year are imputed (see section 2.4.2 below). Missing values are assigned to variables if feasible values cannot be found or inputed.

After all practice settings for a provider have been processed, a single *base* record is created. Included among the base record variables is the provider's UPIN. The UPIN is necessary if the base record is to be merged with other provider records.

2.3 Individual Business Setting Records

Unlike the edits for the base record variables, edits for the business setting variables do not take into consideration values from other practice settings. Suppose, for instance, that a provider practices only in one state. If the state license number is missing on one practice setting but available on another, the missing value is *not* replaced by the available value. Each variable on a business setting record, thus, has either a feasible value or a missing value. A business setting record is created for each practice setting. Appendix B has a complete listing of the program that creates business setting records.

In addition to the edited variables, each business setting record also includes the provider's UPIN and the carrier-assigned Medicare billing number (CPIN). The UPIN is necessary to link business setting records to the provider's base record. If business setting records are going to be merged to the original practice setting records, it is further necessary to create a variable to

uniquely identify each business (practice) setting.[1]

2.4 Base Record Variables

There are two types of base record variables: (1) variables with no edits or minor adjustments to the data and (2) variables subjected to extensive edits.

2.4.1 Unedited Variables

There are two types of unedited variables: (1) name components and (2) sanction information. The name components consist of the provider's last name, first name, middle name (or initial), and suffix (e.g., Jr.). The only adjustment to the name components was to left-justify each. The three sanction variables are not edited. However, algorithms to classify values as feasible, missing, erroneous, dubious, and unknown are provided.

2.4.2 Edited Variables

There are eight edited base record variables: (1) birth year, (2) full date of birth, (3) death year, (4) full date of death, (5) school code, (6) graduation year, (7) credentials, and (8) physician status. Data specifications and editing algorithms are provided for each variable.[2] Since each variable has its own discussion, it is not repeated here. As discussed in section 1.4, advanced programming languages such as SAS provide more programming flexibility than

[1] In the program to merge the file with edited variables back to the original file, the variable is named OBSNUM. Although not OBSNUM is not present in the program in Appendix B, it can be included in the KEEP statement for IN.JUL0293 – OBSNUM is created during the creation of the SAS data set JUL0293.

[2] Included in the edit specifications are values to be assigned to a flag variable. The actual flag assignments, however, are not in the accompanying flowcharts and SAS programs. The flag assignments are provided for those circumstances in which a convenient guide to create a flag variable is needed.

implied by flowcharts. The SAS programs and extracts presented in this chapter, therefore, fulfill the objectives of the flowcharts. The accompanying SAS extracts are presented to help researchers identify the relevant code contained in the Appendices. The SAS code that follows each flowchart is extracted from the full program (Appendix A). These SAS code extracts may not perform their intended edits if used in isolation from the rest of the program.

There are, however, several instances in which inconsistencies seem to arise; and they are discussed here. For instance, in order for birth year to be feasible, all that is necessary is that the year component of the date of birth fall within the minimum and maximum values. However, because both the month and day components of the full date of birth also need to be feasible in order for the edited full date of birth to be feasible, it is possible that the edited birth year value is feasible but that the full date of birth is coded as missing. A similar situation arises with regard to death year and the full date of death.

Because both birth year and graduation year have high number of missing values, imputation algorithms were developed for cases in which one value is feasible but the other is missing. When graduation year is missing, its value depends on the value of birth year (if not missing) and average graduation *age* as follows:

Imputed Graduation Age	**Birth Year**
27	$5 \leq$ birth year ≤ 59
26	$60 \leq$ birth year ≤ 65
25	$66 \leq$ birth year ≤ 69

The imputed value of graduation year then equals the sum of birth year and the imputed value of graduation age. For providers with feasible values of birth year and graduation year, mean graduation ages were determined for each of the above three ranges for birth year. These means are used as the imputed values of graduation age when graduation year is missing. A similar process is used when birth year is missing. Likewise, if birth year is missing but

graduation year is not, graduation age is imputed as follows:

Imputed Graduation Age	Graduation Year
25	30 ≤ graduation year ≤ 49
26	50 ≤ graduation year ≤ 73
27	74 ≤ graduation year ≤ 82
28	83 ≤ graduation year ≤ 93

The imputed value of birth year then equals graduation year minus the imputed value of graduation age. The imputed value of graduation age is carried in the base record when the imputation is performed. If imputation is not performed then graduation age is set to the missing value code. For base records without an imputed graduation age, it is calculated as part of the logical consistency checks between birth year and graduation year; and, if necessary, it is edited as part of the comparisons between graduation year and medical school open/close dates (Chapter 3). Because of the complicated process for creating and editing graduation age, specifications and editing flowcharts are not provided in this chapter.

Unlike most of the other variables in this section, the value of physician status depends entirely on the value of credentials. The *Physician Registry's* specifications were used to create physician status.

2.5 Business Setting Variables

There are two types of business setting variables: (1) variables with no edits or minor adjustments to the data and (2) variables subjected to extensive edits.

2.5.1 Unedited Variables

The unedited variables include the date the business (practice) setting was added to the *Physician Registry* (SETDATE), the carrier number (CARRIER), the carrier-assigned Medicare billing number (CPIN), the provider's billing zip code (BILLZIP), and the provider's business zip code (BIZZIP). There are several reasons why edits are not provided for the zip codes. First, the zip codes in the Registry file come from different years and the post office updates zip codes annually. Second, SAS provides a built-in crosswalk to FIPS state and county codes. The SAS crosswalk may differ among computer installations, depending on when the SAS license is renewed. Third, non-SAS users may have a third-party zip code crosswalk that is not current. If zip codes are to be used for analytical purposes, researchers should perform feasibility checks and edits.

2.5.2 Edited Variables

Of the edited business setting variables, only one – namely, license number – requires additional comment. For the other variables, the accompanying specifications and editing algorithms provide all necessary information. All edits for the other business setting variables conform to *Registry* specifications.

The *Registry* specifies that the license number field be right-justified. If the provider's license number is not known, the field is filled with blanks. Our edits (1) automatically right-justify the license number and (2) for those cases in which all zeroes are found in the field, the field is filled with all blanks. The *Registry* specifies that zeroes should be used to pad the field to the *left* of a valid license number. This edit, however, is not included in the accompanying algorithms and SAS code.

BASE RECORD: UNEDITED DATA ELEMENTS

DATA ELEMENT: SANCTION CODE

Sanction code should be present if available. Of the approximately four thousand entered codes, about 25% are in the unknown category.

Data Format:

PIC X(01) for both carrier record and Registry File.

Flag assignment hierarchy, in descending order of priority:

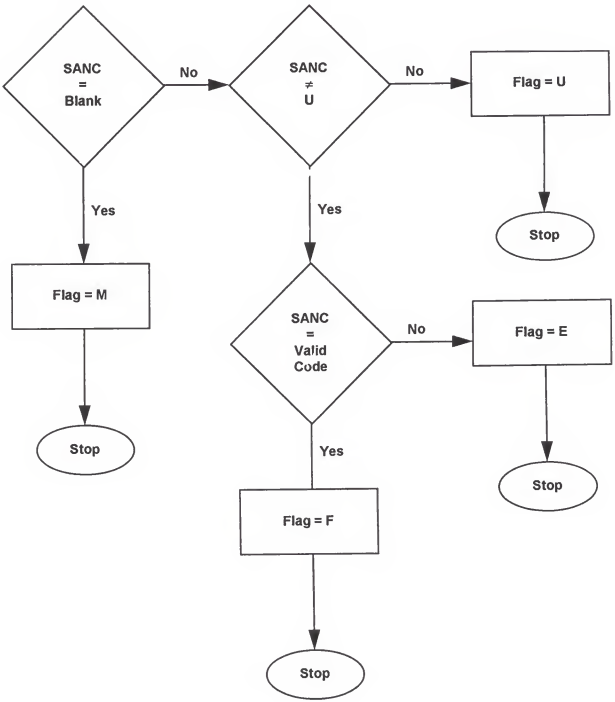
Flag	Definition	Criteria
F	Feasible	A through R
U	Unknown	U
M	Missing (See Note 2)	Blank
E	Erroneous	all other values

Note:

(1) Edits and flagging not actually performed on sanction code in Appendix A.

(2) A blank is used to indicate that a provider is not sanctioned; therefore, the blank value is not considered a missing value. The only time a blank value might be indicative of a problem is when sanction date or length have feasible values; however, this algorithm is not intended to evaluate a blank in relation to data in the sanction date or sanction length fields.

FLOWCHART FOR SANCTION CODE (SANC) INTEGRITY



```
 /*********************************************************/
 /*                                                       */
 /* -CHECK SANCTION CODE                                  */
 /*                                                       */
 /*********************************************************/

DATA S ;
 SET IN.JUL0293
       (KEEP= ESANC1);


 /* SANCTION CODE (ESANC1) */

 IF 'A'<=ESANC1<='R' THEN HSANCS='F';
 ELSE IF ESANC1=' ' THEN HSANCS='M';
 ELSE IF ESANC1='U' THEN HSANCS='U';
 ELSE HSANCS='E';
 OUTPUT S;

RUN;

PROC FREQ DATA=S ORDER=FREQ; TABLES HSANCS;
 TITLE1 '**** SETTING ****';
RUN;
```

DATA ELEMENT: SANCTION DATE

Sanction date should be present if available. Implausible dates frequently occur, especially years prior to the establishment of Medicare or after 1993.

Data Format:

For both Carrier Record and Registry File: PIC 9(04); Component Order: MMYY

Flag assignment hierarchy, in descending order of priority:

Flag	Definition	Criteria
F	Feasible	$66 \leq year \leq 93$ and $1 \leq month \leq 12$
M	Missing (See Note 3)	0 or blank
E	Erroneous	all other values

Notes:

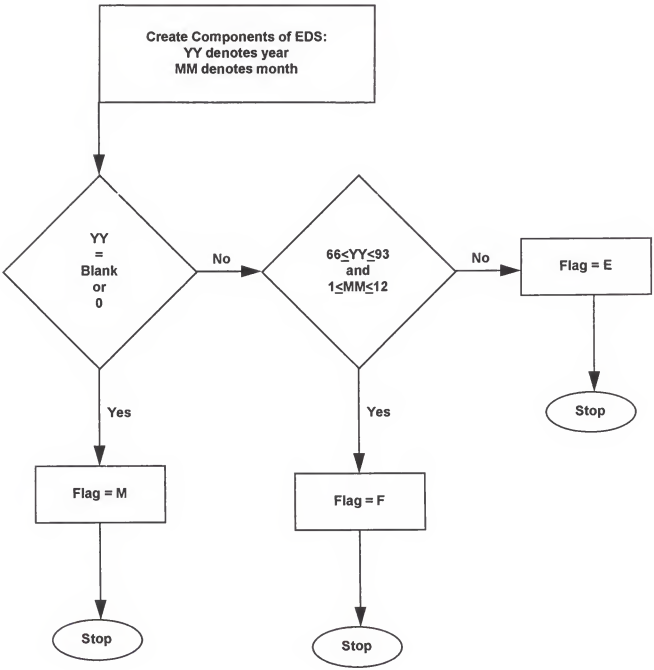
(1) Edits and flagging not actually performed on sanction date in Appencix A.

(2) The upper bound of feasible years needs to be annually incremented upward.

(3) Similar to sanction code, non-sanctioned providers should have a zero or blank, which normally should not be considered a missing value.

(4) If the year component of the date is zero or blank, then the sanction date is flagged as missing, regardless of the contents of the month field.

FLOWCHART FOR EFFECTIVE DATE OF SANCTION (EDS) INTEGRITY



```
  /*******************************************************/
  /*                                                     */
  /* -Check sanction date                                */
  /*                                                     */
  /*******************************************************/

DATA S;
 SET IN.JUL0293
       (KEEP=ESANCDAT);

 /* EFFECTIVE DATE OF SANCTION (ESANCDAT) */

IF ESANCDAT^=. THEN DO;
 SANC1MO = SUBSTR(ESANCDAT,9,2);
 SANC1YR = SUBSTR(ESANCDAT,11,2);
END;

 IF ESANCDAT IN (0,.) THEN HSANCDTS='M';
 ELSE IF ('66'<=SANC1YR<='93') & (' 1'<=SANC1MO<='12') & SANC1MO^='00'
  THEN HSANCDTS='F';
 ELSE HSANCDTS='E';
 OUTPUT S;

RUN;

PROC FREQ DATA=S ORDER=FREQ; TABLES HSANCDTS;
 TITLE1 '**** SETTING ****';
RUN;
```

DATA ELEMENT: SANCTION LENGTH

Sanction length should be present if available. The list of sanctioned providers issued by HCFA's Office of the Inspector General (OIG) includes the sanction term (length) as well as the type of sanction and the effective date. Sanction terms longer than 15 years are incremented by five. Since formal guidelines for coding sanction length have not been issued, rather than denote values such as 16 as erroneous, they have been classified as dubious.

Data Format:

Carrier Record: PIC 9(02)
Registry File: PIC S9(03) COMP-3

Flag assignment hierarchy, in descending order of priority:

Flag	Definition	Criteria
F	Feasible	1-15, 20, 25, 30, 35, 40, 45, 50, or 99
D	Dubious	16-19, 21-24, 26-29, 31-34, 36-39, 41-44, 46-49, 51-98
M	Missing (See Note 3)	0 or blank
E	Erroneous	length $\leq$ 0 or length $\geq$100

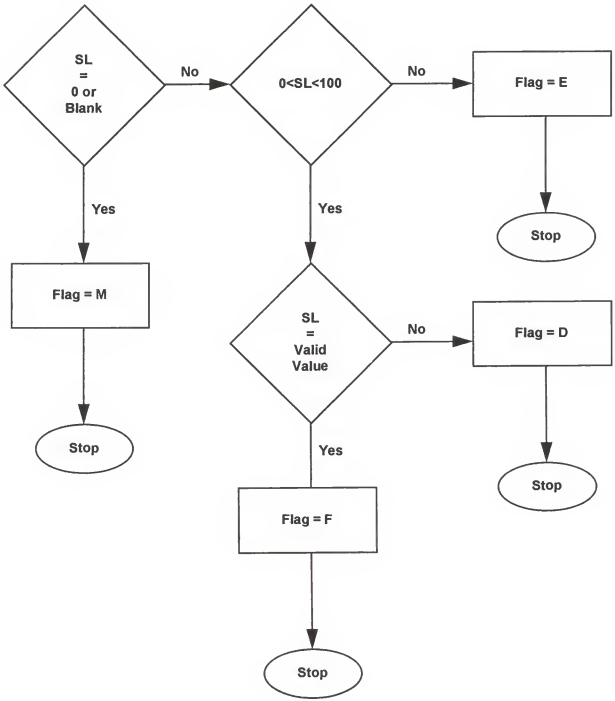
Notes:

(1) Edits and flagging not actually performed on sanction length in Appendix A.

(2) Starting with sanction length 15, sanction length increments by five years. A value of 99 indicates an indefinite sanction length.

(3) As with sanction date, non-sanctioned providers should have a zero or blank and, thus should not be considered a missing value.

FLOWCHART FOR SANCTION LENGTH (SL) INTEGRITY



```
/*********************************************************/
/*                                                       */
/* -SANCTION LENGTH INTEGRITY CHECKS                     */
/*                                                       */
/*********************************************************/

DATA S;
 SET IN.JUL0293
       (KEEP=SANC2YRS);


/* LENGTH OF SANCTION (ESANCYRS) */

IF (1<=ESANCYRS<=15) OR (ESANCYRS IN (20,25,30,35,40,45,50,99))
  THEN HSANCYRS='F';
 ELSE IF ESANCYRS IN (0,.) THEN HSANCYRS='M';
 ELSE IF (16<=ESANCYRS<=19) OR (21<=ESANCYRS<=24) OR
         (26<=ESANCYRS<=29) OR (31<=ESANCYRS<=34) OR
         (36<=ESANCYRS<=39) OR (41<=ESANCYRS<=44) OR
         (46<=ESANCYRS<=49) OR (51<=ESANCYRS<=98)
  THEN HSANCYRS='D';
 ELSE HSANCYRS='E';
 OUTPUT S;

RUN;

PROC FREQ DATA=S ORDER=FREQ; TABLES HSANCYRS;
 TITLE1 '**** SETTING ****';
RUN;
```

BASE RECORD: EDITED DATA ELEMENTS

DATA ELEMENT: BIRTH YEAR

While date of birth is a required Registry data element, carriers are allowed to submit only the year portion if the month and day values cannot be obtained.

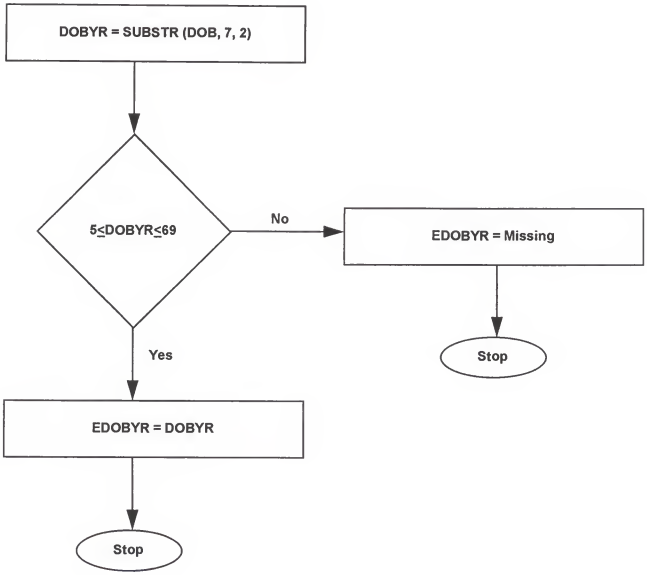
Flag	Definition	Criteria
F	Feasible	$5 \leq Year \leq 69$
M	Missing	all other values

Notes:

(1) See those for date of birth.

(2) In the actual computer program (Appendix A), determination of birth year is integrated with that for the full date of birth.

FLOWCHART FOR
BIRTH YEAR (EDOBYR) EDITS



Notes:

DOB denotes either the header or practice setting version of birth date;
DOBYR denotes birth year.

```
/*************************************************************/
/*                                                           */
/* -RESEARCHER EDITS: BIRTH YEAR ONLY                        */
/*                                                           */
/*************************************************************/


  DOBYR = SUBSTR(DOB,7,2);
  SETDOBYR = SUBSTR(SETDOB,7,2);

  IF ' 5'<=DOBYR<='69' THEN EDOBYR=DOBYR;
    ELSE IF ' 5'<=SETDOBYR<='69' THEN EDOBYR=SETDOBYR;
```

DATA ELEMENT: DATE OF BIRTH

Date of birth is a required data element. Month and day values are sometimes out of range (e.g., month = 15). Year sometimes has implausible values (e.g., 82). Sometimes the value does not correspond to the carrier record format or to the Registry file format (see Exhibit 1-1) as when a date of birth has 5000 as its value instead of its actual value.

Data Format:

Carrier Record: PIC 9(06); Component Order: MMDDYY
Registry File: PIC S9(09) COMP-3; Component Order: YYMMDD

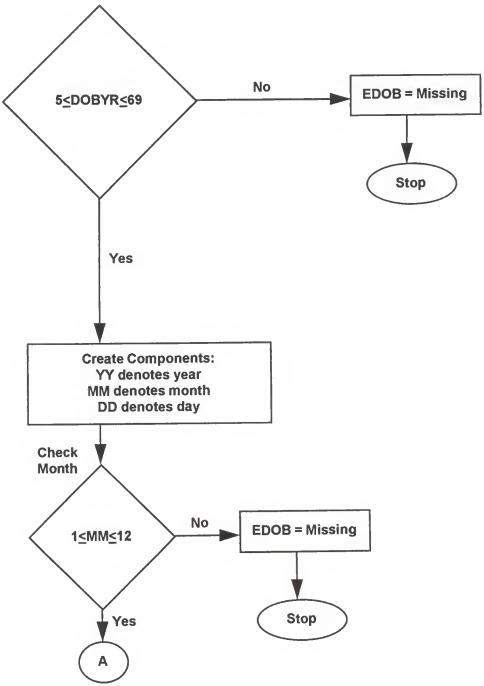
Flag	Definition	Criteria
F	Feasible	5≤year≤69 and 1≤month≤12 and day value consistent with month and year
M	Missing	all other values

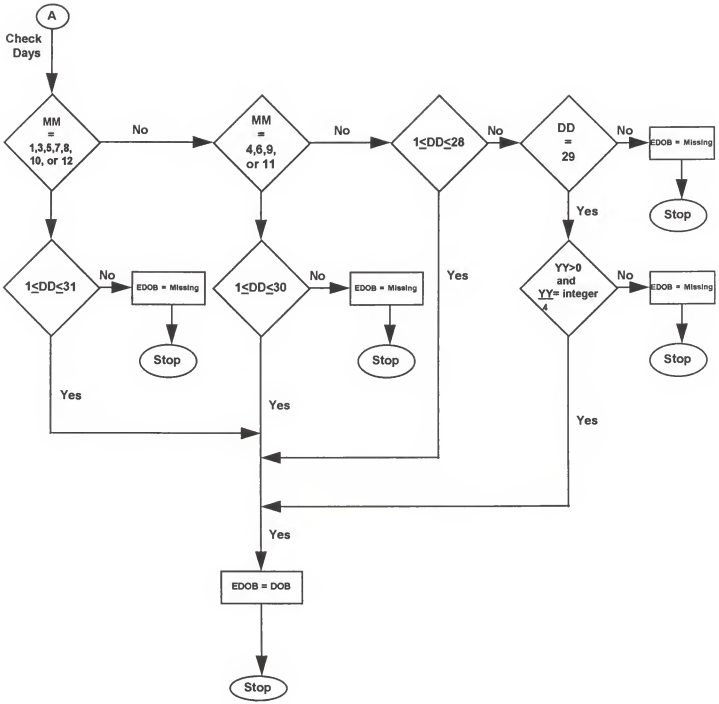
Notes:

(1) Because the month and day values need to be feasible in order for birth date to be feasible, it is possible that the value for birth year (EDOBYR) is in the feasible range while EDOB is not.

(2) Providers are not allowed to be age 23 or younger. Thus, for the July 1993 Registry file, the upper bound for feasible values of year was 69. The upper bound needs to be annually adjusted upward, 70 for 1994, 71 for 1995, and so on. Note that there might be infrequent cases in which the provider actually is 23 years old or less - such instances will be coded as missing or imputed from graduation year because of the probability that birth years later than 1970 are errors.

FLOWCHART FOR DATE OF BIRTH (EDOB) EDITS





Notes:

DOB denotes either the header or practice setting version of birth date.

```
/*************************************************************/
/*                                                           */
/* -RESEARCHER EDITS: DATE OF BIRTH (FULL)                   */
/*                                                           */
/*************************************************************/


  DOBYR = SUBSTR(DOB,7,2);
  DOBMMDD = SUBSTR(DOB,9,4);
  SETDOBYR = SUBSTR(SETDOB,7,2);
  SDOBMMDD = SUBSTR(SETDOB,9,4);

  IF ' 5'<=DOBYR<='69' THEN DO;
    EDOBYR=DOBYR;
    IF ('0101'<=DOBMMDD<='0131') ] ('0201'<=DOBMMDD<='0228') ]
       (DOBMMDD='0229' & (DOBYR/4=INT(DOBYR/4))) ]
       ('0301'<=DOBMMDD<='0331') ] ('0401'<=DOBMMDD<='0430') ]
       ('0501'<=DOBMMDD<='0531') ] ('0601'<=DOBMMDD<='0630') ]
       ('0701'<=DOBMMDD<='0731') ] ('0801'<=DOBMMDD<='0831') ]
       ('0901'<=DOBMMDD<='0930') ] ('1001'<=DOBMMDD<='1031') ]
       ('1101'<=DOBMMDD<='1130') ] ('1201'<=DOBMMDD<='1231')
       THEN EDOB=DOB;
  END;

  IF EDOB=. THEN DO;
   IF ' 5'<=SETDOBYR<='69' THEN DO;
    EDOBYR=SETDOBYR;
    IF ('0101'<=SDOBMMDD<='0131') ] ('0201'<=SDOBMMDD<='0228') ]
       (SDOBMMDD='0229' & (SETDOBYR/4=INT(SETDOBYR/4))) ]
       ('0301'<=SDOBMMDD<='0331') ] ('0401'<=SDOBMMDD<='0430') ]
       ('0501'<=SDOBMMDD<='0531') ] ('0601'<=SDOBMMDD<='0630') ]
       ('0701'<=SDOBMMDD<='0731') ] ('0801'<=SDOBMMDD<='0831') ]
       ('0901'<=SDOBMMDD<='0930') ] ('1001'<=SDOBMMDD<='1031') ]
       ('1101'<=SDOBMMDD<='1130') ] ('1201'<=SDOBMMDD<='1231')
       THEN EDOB=SETDOB;
   END;
  END;
```

DATA ELEMENT: DEATH YEAR

Carriers are allowed to submit only the year portion if the month and day values cannot be obtained.

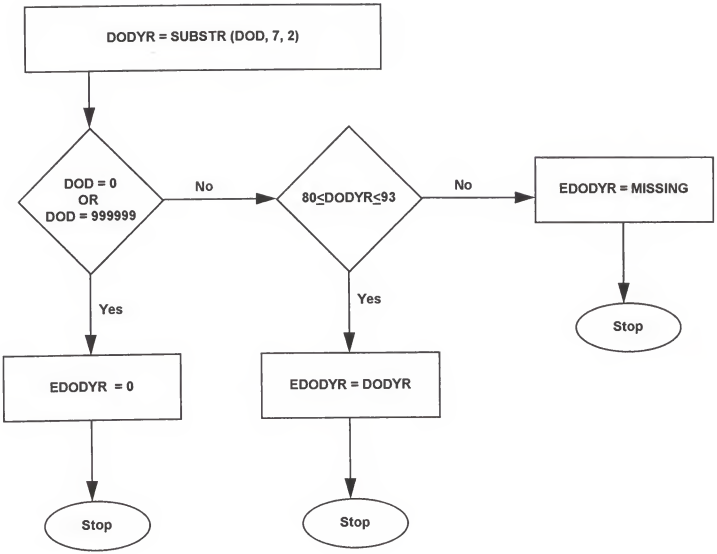
Flag	Definition	Criteria
F	Feasible	80≤Year≤93
0	Alive (See Note 1)	DOD = 0 or 999999
M	Missing	all other values

Notes:

(1) See notes for date of death.

(2) In the actual computer program (Appendix A), determination of death year is integrated with that for the full date of death.

FLOWCHART FOR DEATH YEAR (EDODYR) EDITS



Notes:

DOD denotes the header version of death date;
DODYR denotes death year.

```
/*************************************************************/
/*                                                           */
/* -RESEARCHER EDITS: DEATH YEAR ONLY                        */
/*                                                           */
/*************************************************************/


DODYR = SUBSTR(DOD,7,2);

IF DOD=999999 OR DOD=0 THEN EDODYR=' 0';
 ELSE IF DODYR IN ('80','81','82','83','84','85','86','87'
                   '88','89','90','91','92','93') THEN EDODYR=DODYR;
```

DATA ELEMENT: DATE OF DEATH

The month and day components of date of death are often missing (set to zero) or have out-of-range values. A zero indicates that the provider is alive (as far as the carrier knows).

Data Format:

Carrier Record: PIC 9(06); Component Order: MMDDYY
Registry File: PIC S9(09) COMP-3; Component Order: YYMMDD

Flag	Definition	Criteria
F	Feasible	80≤Year≤93 and 1≤Month≤12 and day value consistent with month and day
0	Alive	DOD = 0 or 999999
M	Missing	all other values

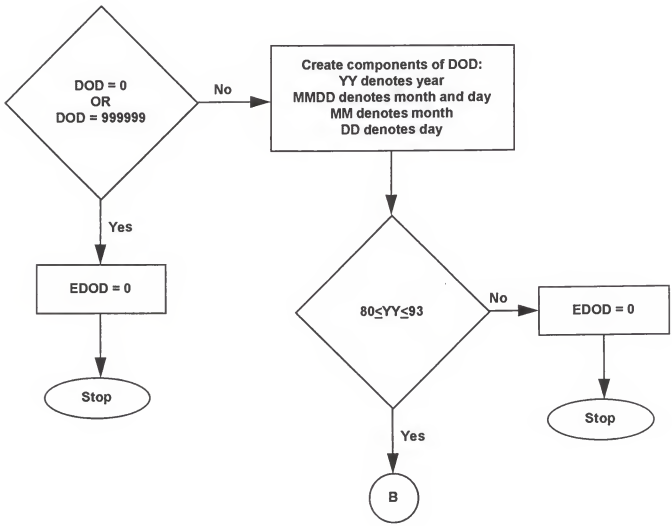
Notes:

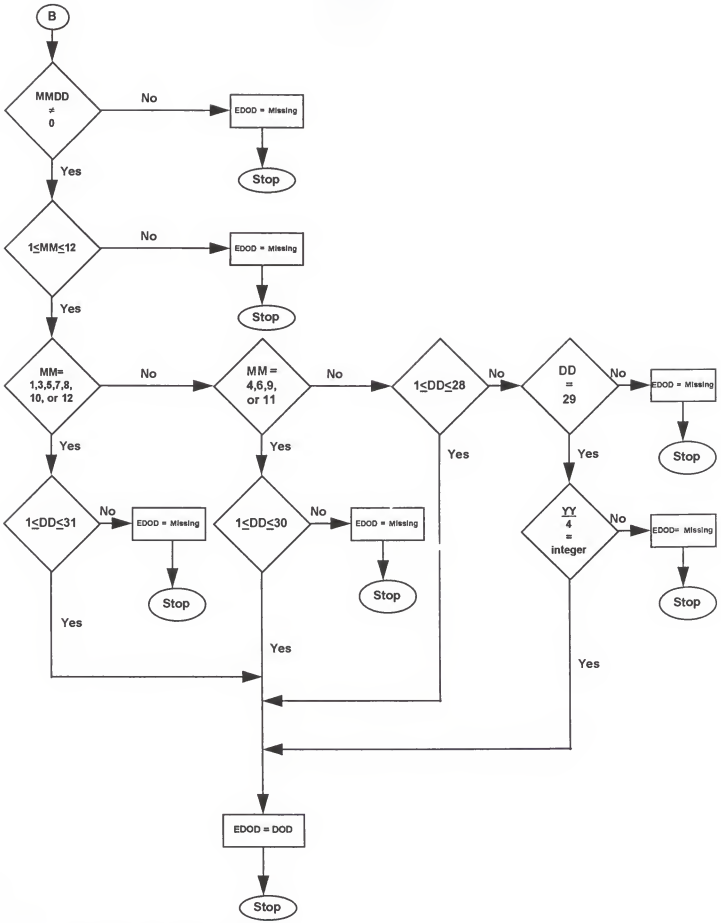
(1) Because the month and day values need to be feasible in order for the death date to be feasible, it is possible that the value for death year (EDODYR) is in the feasible range while EDOD is not.

(2) There are several UPINs which have all nines (9s) in the field. This appears to be a mistake and that it was intended to zero-fill the field.

(3) During the implementation phase carriers were instructed to only submit practice settings that had claim activity during the two preceding years (1987 and 1988). Some carriers, however, sent their entire provider file. A lower bound for death year of 80 was chosen instead of 1987 in order to detect inappropriate use of UPINs assigned to deceased providers.

FLOWCHART FOR
DATE OF DEATH (EDOD) EDITS





Notes:

DOD denotes header version of death date.

```
/*****************************************************************/
/*                                                               */
/* -RESEARCHER EDITS: DATE OF DEATH (FULL)                       */
/*                                                               */
/*****************************************************************/


DODYR = SUBSTR(DOD,7,2);
DODMMDD = SUBSTR(DOD,9,4);

IF DOD=999999 OR DOD=0 THEN DO;
 EDODYR=' 0';
 EDOD=0;
END;
 ELSE IF DODYR IN ('80','81','82','83','84','85','86','87'
                   '88','89','90','91','92','93') THEN DO;
   EDODYR=DODYR;
   IF ('80'<=DODYR<='92' &
    (('0101'<=DODMMDD<='0131') ] ('0201'<=DODMMDD<='0228') ]
    (DODMMDD='0229' & (DODYR/4=INT(DODYR/4))) ]
    ('0301'<=DODMMDD<='0331') ] ('0401'<=DODMMDD<='0430') ]
    ('0501'<=DODMMDD<='0531') ] ('0601'<=DODMMDD<='0630') ]
    ('0701'<=DODMMDD<='0731') ] ('0801'<=DODMMDD<='0831') ]
    ('0901'<=DODMMDD<='0930') ] ('1001'<=DODMMDD<='1031') ]
    ('1101'<=DODMMDD<='1130') ] ('1201'<=DODMMDD<='1231'))) ]
    (DODYR='93' &
    (('0101'<=DODMMDD<='0131') ] ('0201'<=DODMMDD<='0229') ]
    ('0301'<=DODMMDD<='0331') ] ('0401'<=DODMMDD<='0430') ]
    ('0501'<=DODMMDD<='0531') ] ('0601'<=DODMMDD<='0630')))
    THEN EDOD=DOD;
  END;
```

DATA ELEMENT: SCHOOL CODE

School code is a required data element. Values equal to all blanks frequently occur. Completely erroneous codes occur less frequently. Since the establishment of the Registry, some codes have been superceded (retired).

Data Format:

PIC X (05) for both carrier record and Registry File.

Flag	Definition	Criteria
F	Feasible	codes present in Exhibit 2-1
M	Missing	all other values

Notes:

(1) School codes were not required for the initial Registry load.

(2) Near the beginning of the implementation phase of the Registry, carriers were instructed to use the following codes for limited license practitioners (*Program Memorandum*, Carrier, Transmittal No. B-89-2, February 1989):

Code	Credential
10000	Doctor of Dental Medicine (DDM)
20000	Doctor of Dental Surgery (DDS)
30000	Podiatry (DPM)
40000	Optometry (OD)
50000	Chiropractic (CH)

(a) In October 1989, individual school codes were listed in Exhibits 3B-3D of the *Medicare Carriers Manual*, Part 4, Professional Relations, Transmittal No. 2 (October 1989).

(b) Additionally, in Transmittal No. 2 (Exhibit 3C), code 40000 was assigned to the School of Optometry at the University of Alabama in Birmingham.

(3) Podiatric school codes were revised in 1991 (Exhibit 3B, *Medicare Carriers Manual*, Part 4, Professional Relations, Transmittal No. 6, December 1991):

Old	New	Name
000XA	30000	California College of Podiatric Medicine
000XB	30100	Dr. William M. Scholl College of Podiatric Medicine
000XC	30200	New York College of Podiatric Medicine
000XD	30300	Ohio College of Podiatric Medicine
000XE	30400	Pennsylvania College of Podiatric Medicine
000XK	30500	Barry University School of Podiatric Medicine
000XJ	30600	College of Podiatric Medicine and Surgery, University of Osteopathic Medicine and Health Sciences

(a) Codes 000XF-000XI have not been superseded and are assigned to podiatric schools that are now closed.
(b) Prior to the above revisions, school code 30000 was used to indicate any podiatry school.

(4) Even though individual codes have been assigned to podiatry, optometry, and chiropactic schools, codes 30000, 40000, and 50000 are excessively used (use of each *catch-all* code constitutes well over half of its respective category's code usage).

EXHIBIT 2-1

REGISTRY SCHOOL CODES

HCFA ID	SCHOOL	LOCATION
00102	University of Alabama School of Medicine	Birmingham, AL
00104	Birmingham Medical College	Birmingham, AL
00106	University of South Alabama College of Medicine	Mobile, AL
00301	University of Arizona College of Medicine	Tuscon, AR
00401	University of Arkansas College of Medicine	Little Rock, AR
00402	College of Physicians and Surgeons	Little Rock, AR
00501	Cooper Medical College	San Francisco, CA
00502	University of California, San Francisco, School of Medicine	San Francisco, CA
00504	California Eclectic Medical College	Los Angeles, CA
00505	Hahneman Medical College of the Pacific (Homeopathic)	San Francisco, CA
00506	University of Southern California School of Medicine	Los Angeles, CA
00507	College of Physicians and Surgeons of San Francisco	San Francisco, CA
00508	Oakland College of Medicine and Surgery	Oakland, CA
00509	College of Physicians and Surgeons of Los Angeles	Los Angeles, CA
00511	Stanford University School of Medicine	Palo Alto, CA
00512	Loma Linda University School of Medicine	Loma Linda, CA
00513	Pacific Medical College	Los Angeles, CA
00514	University of California, Los Angeles, School of Medicine	Los Angeles, CA
00515	University of California, Irvine, California College of Medicine	Irvine, CA
00516	University California Ca College of Med Irvine CA	Irvine, CA
00517	University Of Cal College of Medicine Irvine CA	Irvine, CA
00518	University of California, San Diego, School of Medicine	La Jolla, CA
00519	University of California, Davis, School of Medicine	Davis, CA
00576	College of Osteopathic Medicine of the Pacific	Pomona, CA
00702	University of Colorado School of Medicine	Denver, CO
00705	Denver and Gross College of Medicine	Denver, CO
00801	Yale University School of Medicine	New Haven, CT
00802	University of Connecticut School of Medicine	Farmington, CT
01001	George Washington University School of Medicine and Health Sciences	Washington, D.C.
01002	Georgetown University School of Medicine	Washington, D.C.
01003	Howard University College of Medicine	Washington, D.C.
01102	University of Miami School of Medicine	Miami, FL
01103	University of Florida College of Medicine	Gainesville, FL
01104	University of South Florida College of Medicine	Tampa, FL
01175	Southeastern University of Health Sciences College of Osteopathic Medicine	Miami, FL
01201	Medical College of Georgia	Augusta, GA
01205	Emory University School of Medicine	Augusta, GA
01209	Georgia College of Eclectic Medicine and Surgery	Augusta, GA
01211	Atlanta College of Physicians and Surgeons	Atlanta, GA

EXHIBIT 2-1 (continued)

REGISTRY SCHOOL CODES

HCFA ID	SCHOOL	LOCATION
01212	Atlanta School of Medicine	Atlanta, GA
01218	Hospital Medical College Eclectic	Atlanta, GA
01219	Southern College of Medicine & Surgery	Atlanta, GA
01221	Morehouse School of Medicine	Atlanta, GA
01222	Mercer University School of Medicine	Macon, GA
01401	University of Hawaii John A. Burns School of Medicine	Honolulu, HI
01601	Rush Medical College	Chicago, IL
01602	University of Chicago Pritzker School of Medicine	Chicago, IL
01604	Hahneman Medical College and Hospital	Chicago, IL
01605	College Of Medicine and Surgery (Physio-Medical)	Chicago, IL
01606	Northwestern University Medical School	Chicago, IL
01608	Bennett Medical College	Chicago, IL
01609	Northwestern University Womans Medical School	Chicago, IL
01610	The Chicago Homeopathic Medical College	Chicago, IL
01611	University of Illinois College of Medicine	Chicago, IL
01613	Harvey Medical College	Chicago, IL
01614	National Medical University	Chicago, IL
01615	Hering Medical College (Homeopathic)	Chicago, IL
01616	Jenner Medical College	Chicago, IL
01617	Illinois Medical College	Chicago, IL
01618	Dunham Medical College	Chicago, IL
01619	American Medical Missionary College	Battle Creek, MI & Chicago, IL
01622	Chicago College of Medicine and Surgery	Chicago, IL
01623	Dearborn Medical College	Chicago, IL
01642	University of Health Sciences/The Chicago Medical School	North Chicago, IL
01643	Loyola University of Chicago, Stritch School of Medicine	Maywood, IL
01644	The General Medical College	Chicago, IL
01645	Southern Illinois University School of Medicine	Springfield, IL
01675	Chicago College of Osteopathic Medicine	Chicago, IL
01705	Physio-Medical College of Indiana	Indianapolis, IN
01708	The Medical College of Indiana	Indianapolis, IN
01709	Central College of Physicians & Surgeons	Indianapolis, IN
01717	Eclectic Medical College of Indiana	Indianapolis, IN
01718	Indiana Medical College, School of Medicine of Purdue University	Indianapolis, IN
01720	Indiana University School of Medicine	Indianapolis, IN
01801	College of Physicians and Surgeons	Keokuk, IA
01803	University of Iowa College of Medicine	Iowa City, IA
01804	State University of Iowa, College of Homeopathic Medicine	Iowa City, IA
01806	Drake University College of Medicine	Des Moines, IA

EXHIBIT 2-1 (continued)

REGISTRY SCHOOL CODES

riCFA ID	SCHOOL	LOCATION
01808	Sioux City College of Medicine	Sioux City, IA
01810	Keokuk Medical College, College of Physicians and Surgeons	Keokuk, IA
01875	University of Osteopathic Medicine and Health Sciences/College of Osteopathic Medicine and Surg	Des Moines, IA
01902	University of Kansas School of Medicine	Kansas City, KS
01903	Kansas Medical College (Medical Department of Washburn College)	Topeka, KS
01904	College of Physicians and Surgeons of Kansas City	Kansas City, KS
01907	Western Eclactic College of Medicine and Surgery	Kansas City, KS
02001	Kentucky School of Medicine	Louisville, KY
02002	University Of Louisville School of Medicine	Louisville, KY
02004	Louisville Medical College	Louisville, KY
02005	Hospital College of Medicine (Med Dept of Central University)	Louisville, KY
02007	Louisville National Medical College (Med Dept State University)(Black)	Louisville, KY
02008	Southwestern Homeopathic Medical College and Hospital	Louisville, KY
02009	Kentucky University Medical Department	Louisville, KY
02011	Louisville and Hospital Medical College	Louisville, KY
02012	University of Kentucky College of Medicine	Lexington, KY
02101	Tulane University School of Medicine	New Orleans, LA
02104	Flint Medical College of New Orleans University (Black)	New Orleans, LA
02105	Louisiana State University School of Medicine in New Orleans	New Orleans, LA
02106	Louisiana State University School of Medicine in Shreveport	Shreveport, LA
02201	Bowdoin Medical School	Brunswick and Portland, ME
02275	University of New England College of Osteopathic Medicine	Biddeford, ME
02301	University of Maryland School of Medicine	Baltimore, MD
02303	College of Physicians and Surgeons of Baltimore	Baltimore, MD
02304	Baltimore Medical College	Baltimore, MD
02305	Woman's Medical College of Baltimore	Baltimore, MD
02306	Baltimore University School of Medicine	Baltimore, MD
02307	John Hopkins University School of Medicine	Baltimore, MD
02308	Atlantic Medical College	Baltimore, MD
02309	Maryland Medical College	Baltimore, MD
02311	Maryland College of Eclectic Medicine and Surgery	Baltimore, MD
02312	Uniformed Services University of Health Science	Bethesda, MD
02401	Harvard Medical School	Boston, MA
02405	Boston University School of Medicine	Boston, MA
02406	College of Physicians and Surgeons	Boston, MA
02407	Tufts University School of Medicine	Boston, MA
02415	Middlesex University School of Medicine	Waltham, MA
02416	University of Massachusetts Medical School	Worcester, MA
02501	University of Michigan Medical School	Ann Arbor, MI

EXHIBIT 2-1 (continued)

REGISTRY SCHOOL CODES

HCFA ID	SCHOOL	LOCATION
02505	University of Michigan Homeopathic Medical School	Ann Arbor, MI
02507	Wayne State University School of Medicine	Detroit, MI
02508	Michigan College of Medicine & Surgery	Detroit, MI
02512	Michigan State University College of Human Medicine	East Lansing, MI
02576	Michigan State University College of Osteopathic Medicine	East Lansing, MI
02604	University of Minnesota Medical School	Minneapolis, MN
02605	Minneapolis College of Physicians and Surgeons	Minneapolis, MN
02607	University of Minnesota Duluth School of Medicine	Duluth, MN
02608	Mayo Medical School	Rochester, MN
02701	University Mississippi School of Medicine	Jackson, MI
02702	Mississippi Medical College	Meridian, MI
02801	Missouri Medical College	St. Louis, MO
02802	Washington University Medical School	St. Louis, MO
02803	University of Missouri Columbia School of Medicine	Columbia, MO
02805	Homeopathic Medical College of Missouri	St. Louis, MO
02807	St. Louis College of Physicians and Surgeons	St. Louis, MO
02808	Kansas City Medical College	Kansas City, MO
02810	National University of the Arts and Sciences Medical Department	St. Louis, MO
02820	University Medical College of Kansas City	Kansas City, MO
02822	Ensworth Medical College	St. Joseph, MO
02826	Kansas City Homeopathic Medical College	Kansas City, MO
02828	Barnes Medical College	St. Louis, MO
02830	Woman's Medical College	Kansas City, MO
02833	Eclectic Medical University	Kansas City, MO
02834	Saint Louis University School of Medicine	St. Louis, MO
02835	Southwest School of Medicine and Hospital	Kansas City, MO
02843	Kansas City College of Medicine and Surgery	Kansas City, MO
02844	Kansas City University of Physicians and Surgeons	Kansas City, MO
02845	Mid West Medical College	Kansas City, MO
02846	University of Missouri Kansas City School of Medicine	Kansas City, MO
02878	The University of Health Sciences College of Osteopathic Medicine	Kansas City, MO
02879	Kirksville College of Osteopathic Medicine	Kirksville, MO
03004	Lincoln Medical College, Eclectic	Lincoln, NE
03005	University of Nebraska College of Medicine	Omaha, NE
03006	Creighton University School of Medicine	Omaha, NE
03007	Nebraska College of Medicine	Lincoln, NE
03101	University of Nevada School of Medicine	Reno, NV
03201	Dartmouth Medical School	Hanover, NH
03305	University of Medicine and Dentistry of New Jersey-New Jersey Medical School	Newark, NJ

EXHIBIT 2-1 (continued)

REGISTRY SCHOOL CODES

HCFA ID	SCHOOL	LOCATION
03306	University of Medicine and Dentistry of New Jersey- Robert Wood Johnson Medical School	Piscataway, NJ
03375	University of Medicine and Dentistry of New Jersey-New Jersey School of Osteopathic Medicine	Camden, NJ
03401	University of New Mexico School of Medicine	Albuquerque, NM
03501	Columbia University College of Physicians and Surgeons	New York, NY
03503	Albany Medical College of Union University	Albany, NY
03506	State University of New York at Buffalo School of Medicine	Buffalo, NY
03508	State University of New York Health Science Center at Brooklyn	Brooklyn, NY
03509	New York Medical College	Valhalla, NY
03510	Bellevue Hospital Medical College	New York, NY
03511	New York Medical College and Hospital for Women, Homeopathic	New York, NY
03513	Eclectic Medical College of the City of New York	New York, NY
03515	State University of New York Health Science Center	Syracuse, NY
03519	New York University School of Medicine	New York, NY
03520	Cornell University Medical College	New York, NY
03543	Fordham University School of Medicine	New York, NY
03545	University of Rochester School of Medicine and Dentistry	Rochester, NY
03546	Albert Einstein College of Medicine of Yeshiva University	New York, NY
03547	Mount Sinai School of Medicine of the City University of New York	New York, NY
03548	State University of New York at Stony Brook School of Medicine	Stony Brook, NY
03575	New York College of Osteopathic Medicine of New York Institute of Technology	Old Westbury, NY
03601	University of North Carolina at Chapel Hill School of Medicine	Chapel Hill, NC
03603	Leonard Medical School (Black)	Raleigh, NC
03604	North Carolina Medical College	Charlotte, NC
03605	Bowman Gray School of Medicine of Wake Forest University	Winston-Salem, NC
03607	Duke University School of Medicine	Durham, NC
03608	East Carolina University School of Medicine	Greenville, NC
03701	University of North Dakota School of Medicine	Grand Forks, ND
03801	Medical College of Ohio	Cincinnati, OH
03802	Eclectic Medical College	Cincinnati, OH
03803	Starling Medical College	Columbus, OH
03806	Case Western Reserve University School of Medicine	Cleveland, OH
03808	Cincinnati College of Medicine and Surgery	Cincinnati, OH
03809	Miami Medical College	Cincinnati, OH
03811	University of Wooster Medical Department	Cleveland, OH
03819	Toledo Medical College (Med Dept of Toledo University)	Toledo, OH
03823	Cleveland Medical College, Homeopathic	Cleveland, OH
03825	Ohio Medical University	Columbus, OH
03826	Cleveland-Pulte Medical College	Cleveland, OH
03840	Ohio State University College of Medicine	Columbus, OH

EXHIBIT 2-1 (continued)

REGISTRY SCHOOL CODES

HCFA ID	SCHOOL	LOCATION
03841	University of Cincinnati College of Medicine	Cincinnati, OH
03843	Medical College of Ohio at Toledo	Toledo, OH
03844	Northeastern Ohio Universities College of Medicine	Rootstown, OH
03845	Wright State University School of Medicine	Dayton, OH
03875	Ohio University College of Osteopathic Medicine	Athens, OH
03901	University of Oklahoma College of Medicine	Oklahoma City, OK
03905	Oral Roberts University School of Medicine	Tulsa, OK
03979	Oklahoma State University College of Osteopathic Medicine	Tulsa, OK
04001	Willamette University Medical Department	Salem, OR
04002	Oregon Health Sciences University School of Medicine	Portland, OR
04101	University of Pennsylvania School of Medicine	Philadelphia, PA
04102	Jefferson Medical College of Thomas Jefferson University	Philadelphia, PA
04107	Medical College of Pennsylvania	Philadelphia, PA
04109	Hahnemann University School of Medicine	Philadelphia, PA
04111	Medico-Chirurgical College of Philadelphia	Philadelphia, PA
04112	University of Pittsburgh School of Medicine	Pittsburgh, PA
04113	Temple University School of Medicine	Philadelphia, PA
04114	Pennsylvania State University College of Medicine	Hershey, PA
04177	Philadelphia College of Osteopathic Medicine	Philadelphia, PA
04201	University of Puerto Rico School of Medicine	San Juan, Puerto Rico
04202	Ponce School of Medicine	Ponce, Puerto Rico
04203	University Central del Caribe Escuela de Medicina	Cayey, Puerto Rico
04204	Ciencias Medicina de San Juan Bautista	Hato Rey, Puerto Rico
04301	Brown University Program in Medicine	Providence, RI
04501	Medical University of South Carolina College of Medicine	Charleston, SC
04504	University of South Carolina School of Medicine	Columbia, SC
04601	University of South Dakota School of Medicine	Vermillion, SD
04701	University of Nashville Medical Department	Nashville, TN
04705	Vanderbilt University School of Medicine	Nashville, TN
04706	University of Tennessee College of Medicine	Memphis, TN
04707	Meharry Medical College School of Medicine	Nashville, TN
04708	Memphis Hospital Medical College	Memphis, TN
04709	Chattanooga Medical College	Chattanooga, TN
04710	Lincoln Memorial University Medical Department	Knoxville, TN
04711	University of the South Medical Department	Sewanee, TN
04713	Knoxville Medical College (Black)	Knoxville, TN
04714	University of West Tennesee College of Medicine and Surgery (Black)	Memphis, TN
04715	College of Physicians and Surgeons	Memphis, TN
04720	East Tennessee State University James H. Dishner College of Medicine	Johnson City, TN

EXHIBIT 2-1 (continued)

REGISTRY SCHOOL CODES

HCFA ID	SCHOOL	LOCATION
04802	University of Texas Medical Branch at Galveston	Galveston, TX
04803	Fort Worth School Of Medicine	Fort Worth, TX
04804	Baylor University College Of Medicine	Houston, TX
04805	Physio-Medical College of Texas	Dallas, TX
04806	Southern Methodist University Medical Department	Dallas, TX
04807	Gate City Medical College	Dallas, TX
04812	University of Texas Southwestern Medical Center at Dallas School of Medicine	Dallas, TX
04813	University of Texas Medical School at San Antonio	San Antonio, TX
04814	University of Texas Medical School at Houston	Houston, TX
04815	Texas Tech University Health Sciences Center School of Medicine	Lubbock, TX
04816	Texas A&M University College of Medicine	College Station, TX
04878	Texas College of Osteopathic Medicine	Fort Worth, TX
04901	University of Utah School of Medicine	Salt Lake City, UT
05002	University of Vermont College of Medicine	Burlington, VT
05101	University of Virginia School of Medicine	Charlottesville, VA
05104	Virginia Commonwealth University Medical College of Virginia	Richmond, VA
05106	University College of Medicine	Richmond, VA
05107	Eastern Virginia Medical School	Norfolk, VA
05404	University of Washington School of Medicine	Seattle, WA
05415	Washington College of Physicians and Surgeons	Seattle, WA
05501	West Virginia University School of Medicine	Morgantown, WV
05502	Marshall University School of Medicine	Huntington, WV
05575	West Virginia School of Osteopathic Medicine	Lewisburg, WV
05602	Wisconsin College of Physicians and Surgeons	Milwaukee, WI
05603	Milwaukee Medical College	Milwaukee, WI
05605	University of Wisconsin Medical School	Madison, WI
05606	Medical College of Wisconsin	Milwaukee, WI
000Xa	California College of Podiatric Medicine	San Francisco, CA
000Xb	Dr. William M. Scholl College of Podiatric Medicine	Chicago, IL
000Xc	New York College of Podiatric Medicine	New York, NY
000Xd	Ohio College of Podiatric Medicine	Cleveland, OH
000Xe	Pennsylvania College of Podiatric Medicine	Philadelphia, PA
000Xk	Barry University School of Podiatric Medicine	Miami Shores, FL
00CXj	College of Podiatric Medicine And Surgery	Des Moines, IA
000Xf	Temple University	Philadelphia, PA
000Xg	Chicago College of Chiropody	Chicago, IL
000Xh	Long Island University College of Podiatry	New York, NY
000Xi	First Institute Podiatry	New York, NY
30000	California College of Podiatric Medicine	San Francisco, CA

EXHIBIT 2-1 (continued)

REGISTRY SCHOOL CODES

HCFA ID	SCHOOL	LOCATION
30100	Dr. William M. Scholl College of Podiatric Medicine	Chicago, IL
30200	New York College of Podiatric Medicine	New York, NY
30300	Ohio College of Podiatric Medicine	Cleveland, OH
30400	Pennsylvania College of Podiatric Medicine	Philadelphia, PA
30500	Barry University School of Podiatric Medicine	Miami Shores, FL
30600	College of Podiatric Medicine And Surgery	Des Moines, IA
40000	University of Alabama School of Optometry	Birmingham, AL
40010	University of California School of Optometry	Berkeley, CA
40020	Ferris State College of Optometry	Big Rapids, MI
40030	University Of Houston School of Optometry	Houston, TX
40040	Illinois College of Optometry at Chicago	Chicago, IL
40050	Indiana University School of Optometry	Bloomington, IN
40060	International American University at Puerto Rico School of Optometry	San Juan, Puerto Rico
40070	University of Missouri School of Optometry	St. Louis, MO
40080	University of Montreal School of Optometry	Montreal, Canada
40090	New England College of Optometry	Boston, MA
40100	Berkeley Northeastern State University College of Optometry	Tahlequah, OK
40110	State University of New York School of Optometry	New York, NY
40120	Ohio State University School of Optometry	Columbus, OH
40130	Pacific University School of Optometry	Columbus, OH
40140	Pennsylvania College of Optometry	Philadelphia, PA
40150	Southern California College of Optometry	Fullerton, CA
04160	Southern College of Optometry	Memphis, TN
40170	University of Waterloo School of Optometry	Ontario, Canada
50000	Chiropractic Unknown School	
50010	Adio/Pennsylvania College of Straight Chiropractic	Levittown, PA
50020	Atlantic State/Columbia Chiropractic	Brooklyn, NY
50030	Bebout College of Chiropractic	Indianapolis, IN
50040	Booker T. Washington Chiropractic	Kansas City, MO
50050	Logan College of Chiropractic	Oklahoma City, OK
50060	Los Angeles Chiropractic	Los Angeles, CA
50065	Central States Physiatrics and Chiropractic	Eaton, OH
50070	Canadian Memorial Chiropractic	Toronto, Ontario
50080	Chiropractic Institute of New York	New York, NY
50090	Crisco Chiropractic	Crisco, TX
50100	Cleveland Chiropractic College of Cleveland University--Los Angeles	Los Angeles, CA
50110	Cleveland Chiropractic College	Kansas City, MO
50120	Columbia Chiropratic	Baltimore, MD
50130	Columbia Chiropratic	Alameda, CA

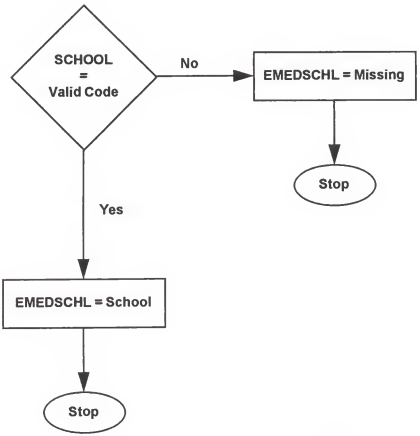
EXHIBIT 2-1 (continued)

REGISTRY SCHOOL CODES

HCFA ID	SCHOOL	LOCATION
50140	Columbia Chiropractic and Naturopathy	Sacramento, CA
50150	Columbia Institute of Chiropratic	New York, NY
50160	Hollywood Chiropractic	Hollywood, CA
50170	Lafayette Chiropractic Institute	Philadelphia, PA
50180	Life Chiropractic College-West	San Lorenzo, CA
50190	Lincoln Chiropractic	Marietta, GA
50200	Reaver School Chiropractic	Dayton, OH
50210	Restview Chiropractic	Seattle, WA
50220	San Francisco Chiropractic	San Francisco, CA
50230	Sherman Straight Chiropractic	Spartanburg, SC
50240	Western States Chiropractic College	Portland, OR
50250	University of Pasadena Chiropractic	Pasadena, CA
50260	Natural Healing Chiropractic	Denver, CO
50270	National College of Chiropractic	Lombard, IL
50280	New York Chiropractic College	Glen Head, NY
50290	Northwestern College of Chiropractic	Bloomington, MN
50300	Palmer College of Chiropractic	Davenport, IA
50310	Palmer College of Chiropractic-West	West Sunnyvale, CA

Note: It appears that three codes have been assigned to the medical school at the University of California at Irvine (see shaded lines on first page of this exhibit).

FLOWCHART FOR
MEDICAL SCHOOL CODE (EMEDSCHL) EDITS



Note:

School denotes either the header or practice setting version of the school code.

```
/*************************************************************/
/*                                                           */
/* -RESEARCHER EDITS: SCHOOL CODE                            */
/*                                                           */
/*************************************************************/


IF MEDSCHL IN ('00102','00104','00106','00301','00401','00402',
               '00501','00502','00504','00505','00506','00507',
               '00508','00509','00511','00512','00513','00514',
               '00515','00516','00517','00518','00519','00576',
               '00702','00705','00801','00802','01001','01002',
               '01003','01102','01103','01104','01175','01201',
               '01205','01209','01211','01212','01218','01219',
               '01221','01222','01401','01601','01602','01604',
               '01605','01606','01608','01609','01610','01611',
               '01613','01614','01615','01616','01617','01618',
               '01619','01622','01623','01642','01643','01644',
               '01645','01675','01705','01708','01709','01717',
               '01718','01720','01801','01803','01804','01806',
               '01808','01810','01875','01902','01903','01904',
               '01907','02001','02002','02004','02005','02007',
               '02008','02009','02011','02012','02101','02104',
               '02105','02106','02201','02275','02301','02303',
               '02304','02305','02306','02307','02308','02309',
               '02311','02312','02401','02405','02406','02407',
               '02415','02416','02501','02505','02507','02508',
               '02512','02576','02604','02605','02607','02608',
               '02701','02702','02801','02802','02803','02805',
               '02807','02808','02810','02820','02822','02826',
               '02828','02830','02833','02834','02835','02843',
               '02844','02845','02846','02878','02879','03004',
               '03005','03006','03007','03101','03201','03305',
               '03306','03375','03401','03501','03503','03506',
               '03508','03509','03510','03511','03513','03515',
               '03519','03520','03543','03545','03546','03547',
               '03548','03575','03601','03603','03604','03605',
               '03607','03608','03701','03801','03802','03803',
               '03806','03808','03809','03811','03819','03823',
               '03825','03826','03840','03841','03843','03844',
               '03845','03875','03901','03905','03979','04001',
               '04002','04101','04102','04107','04109','04111',
               '04112','04113','04114','04177','04201','04202',
               '04203','04204','04301','04501','04504','04601',
               '04701','04705','04706','04707','04708','04709',
               '04710','04711','04713','04714','04715','04720',
               '04802','04803','04804','04805','04806','04807',
               '04812','04813','04814','04815','04816','04878',
               '04901','05002','05101','05104','05106','05107',
               '05404','05415','05501','05502','05575','05602',
               '05603','05605','05606','000XF','000XG',
```

```
                '000XH','000XI','30000','30100','30200','30300',
                '30400','30500','30600','40000','40010','40020',
                '40030','40040','40050','40060','40070','40080',
                '40090','40100','40110','40120','40130','40140',
                '40150','04160','40170','50000','50010','50020',
                '50030','50040','50050','50060','50065','50070',
                '50080','50090','50100','50110','50120','50130',
                '50140','50150','50160','50170','50180','50190',
                '50200','50210','50220','50230','50240','50250',
                '50260','50270','50280','50290','50300','50310',
                '99999','10000','20000',
                '000XA','000XB','000XC','000XD','000XE',
                '000XJ','000XK') THEN EMEDSCHL=MEDSCHL;

ELSE IF SETMDSCH IN ('00102','00104','00106','00301','00401','00402',
                '00501','00502','00504','00505','00506','00507',
                '00508','00509','00511','00512','00513','00514',
                '00515','00516','00517','00518','00519','00576',
                '00702','00705','00801','00802','01001','01002',
                '01003','01102','01103','01104','01175','01201',
                '01205','01209','01211','01212','01218','01219',
                '01221','01222','01401','01601','01602','01604',
                '01605','01606','01608','01609','01610','01611',
                '01613','01614','01615','01616','01617','01618',
                '01619','01622','01623','01642','01643','01644',
                '01645','01675','01705','01708','01709','01717',
                '01718','01720','01801','01803','01804','01806',
                '01808','01810','01875','01902','01903','01904',
                '01907','02001','02002','02004','02005','02007',
                '02008','02009','02011','02012','02101','02104',
                '02105','02106','02201','02275','02301','02303',
                '02304','02305','02306','02307','02308','02309',
                '02311','02312','02401','02405','02406','02407',
                '02415','02416','02501','02505','02507','02508',
                '02512','02576','02604','02605','02607','02608',
                '02701','02702','02801','02802','02803','02805',
                '02807','02808','02810','02820','02822','02826',
                '02828','02830','02833','02834','02835','02843',
                '02844','02845','02846','02878','02879','03004',
                '03005','03006','03007','03101','03201','03305',
                '03306','03375','03401','03501','03503','03506',
                '03508','03509','03510','03511','03513','03515',
                '03519','03520','03543','03545','03546','03547',
                '03548','03575','03601','03603','03604','03605',
                '03607','03608','03701','03801','03802','03803',
                '03806','03808','03809','03811','03819','03823',
                '03825','03826','03840','03841','03843','03844',
                '03845','03875','03901','03905','03979','04001',
                '04002','04101','04102','04107','04109','04111',
                '04112','04113','04114','04177','04201','04202',
                '04203','04204','04301','04501','04504','04601',
                '04701','04705','04706','04707','04708','04709',
```

```
'04710','04711','04713','04714','04715','04720',
'04802','04803','04804','04805','04806','04807',
'04812','04813','04814','04815','04816','04878',
'04901','05002','05101','05104','05106','05107',
'05404','05415','05501','05502','05575','05602',
'05603','05605','05606','000XF','000XG',
'000XH','000XI','30000','30100','30200','30300',
'30400','30500','30600','40000','40010','40020',
'40030','40040','40050','40060','40070','40080',
'40090','40100 ','40110','40120','40130','40140',
'40150','04160','40170','50000','50010','50020',
'50030','50040','50050','50060','50065','50070',
'50080','50090','50100','50110','50120','50130',
'50140','50150','50160','50170','50180','50190',
'50200','50210','50220','50230','50240','50250',
'50260','50270','50280','50290','50300','50310',
'99999','10000','20000',
'000XA','000XB','000XC','000XD','000XE',
'000XJ','000XK') THEN EMEDSCHL=SETMDSCH;
```

DATA ELEMENT: GRADUATION YEAR

Graduation year is a required data element. Values are often missing; year equal to zero frequently occurs. Completely erroneous codes occur less frequently (e.g., year equal to 96).

Data Format:

Carrier Record: PIC 9(02); Component Order: YY
Registry File: PIC S9(09) COMP-3; Component Order: YY

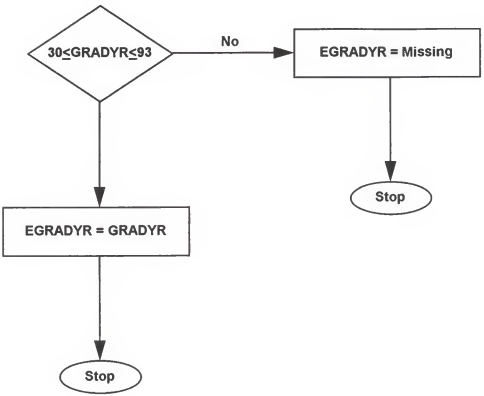
Flag	Definition	Criteria
F	Feasible	30≤graduation year≤93
M	Missing	all other values

Notes:

(1) In 1993, a graduation year of 30 implies an age of 88 (assuming graduation at age 25).

(2) The upper and lower bounds for the feasible criteria should be annually adjusted upward by one.

FLOWCHART FOR GRADUATION YEAR (EGRADYR) EDITS



Note:

GRADYR denotes either the header or practice setting version of graduation year.

```
/*************************************************************/
/*                                                           */
/* -RESEARCHER EDITS: GRADUATION YEAR                        */
/*                                                           */
/*************************************************************/


  IF 30<=GRADYR<=93 THEN EGRADYR=GRADYR;
  ELSE IF 30<=SETGRDYR<=93 THEN EGRADYR=SETGRDYR;
```

DATA ELEMENT: PROVIDER CREDENTIALS

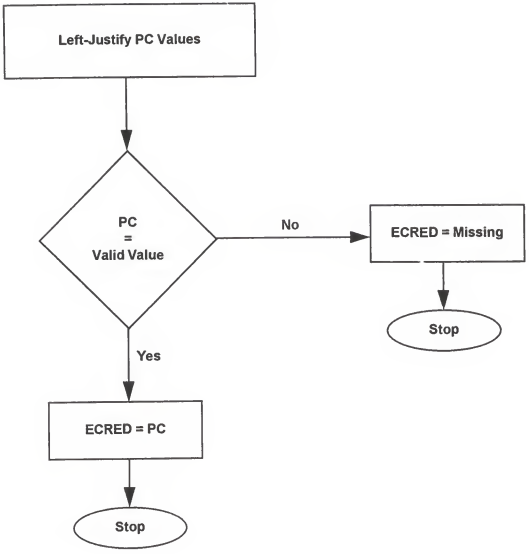
Provider credentials are a required data element and should be left-justified.

Data Format:

PIC X(03) for both Carrier Record and Registry File.

Flag	Definition	Criteria
F	Feasible	MD, DO, DDS, DDM, DPM, CH, or OD
M	Missing	all other values

FLOWCHART FOR PROVIDER CREDENTIALS (ECRED) EDITS



Note:

PC denotes provider credential.

```
/*************************************************************/
/*                                                           */
/* -RESEARCHER EDITS: PROVIDER CREDENTIALS                   */
/*                                                           */
/*************************************************************/


CRED=LEFT(CREDENTL);
IF CRED IN ('MD ','DO ','DDS','DDM','DPM','CH ','OD ') THEN ECRED=CRED;
```

DATA ELEMENT: PHYSICIAN STATUS

Physician status is a required data element.

Data Format:

Carrier Record: PIC 9
Registry File: PIC X(01)

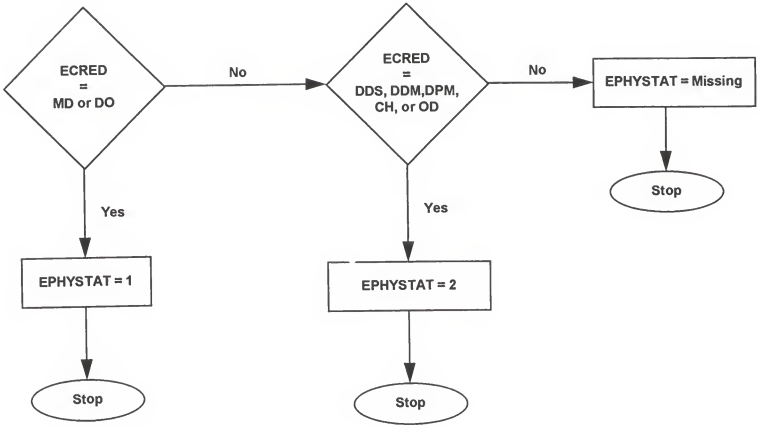
Flag	Definition	Criteria
F	Feasible	1 or 2
M	Missing	all other values

Note:

(1) Physician status is determined by credentials (ECRED):

ECRED	PHYSICIAN STATUS
MD, DO	1
DDS, DDM, DPM, CH, OD	2
All Other	Missing

FLOWCHART FOR PHYSICIAN STATUS (EPHYSTAT) EDITS



Note:

ECRED denotes edited provider credential.

```
/*************************************************************/
/*                                                           */
/* -RESEARCHER EDITS: PHYSICIAN STATUS                       */
/*                                                           */
/*************************************************************/


IF ECRED IN ('MD ','DO ') THEN EPHYSTAT=1;
 ELSE IF ECRED IN ('DDS','DDM','DPM','CH ','OD ') THEN EPHYSTAT=2;
```

BUSINESS SETTING RECORD: EDITED DATA ELEMENTS

DATA ELEMENT: STATE LICENSE NUMBER

State license number is a required data element. It should be right-justified and preceded (padded) with zeroes. Left justification and filling with zeroes instead of blanks are the two most frequent types of non-conformance with Registry specifications.

Data Format:

PIC X(12) for both Carrier Record and Registry File.

Flag assignment hierarchy, in descending order of priority:

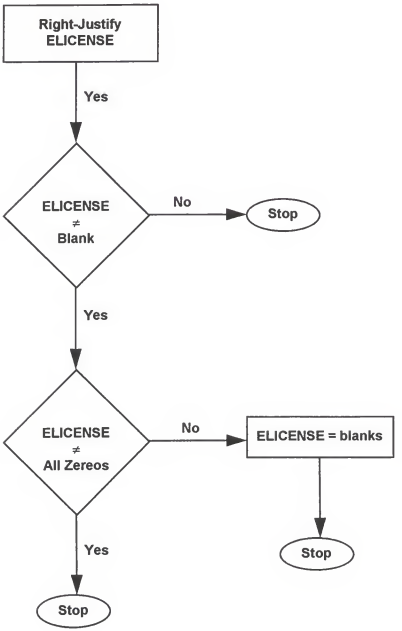
Flag	Definition	Criteria
M	Missing	zeroes or blanks in all byte positions
F	Feasible	all other values

Notes:

(1) Feasible in this instance means that the data element conforms to Registry specification; however, it does not mean that the actual value is correct, or for that matter, conforms to the issuing state's license number format.

(2) Although required now, state license number was not required for the initial Registry load.

FLOWCHART FOR
STATE LICENSE NUMBER (ELICENSE) EDITS



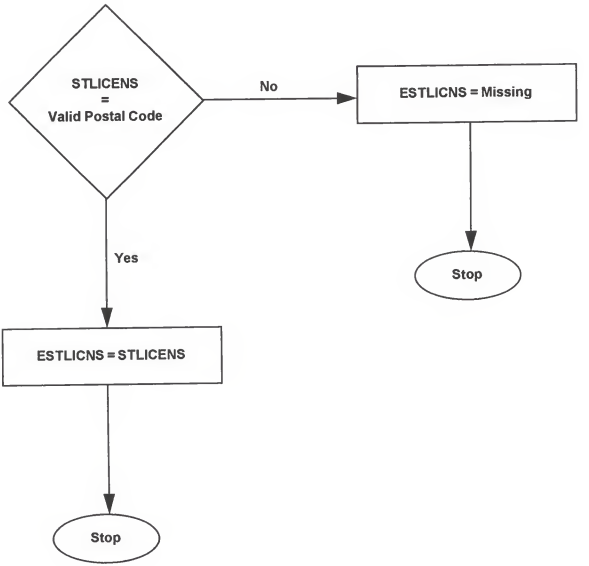
```
/*************************************************************/
/*                                                           */
/* -RESEARCHER EDITS: STATE LICENSE NUMBER                   */
/*                                                           */
/*************************************************************/


IF LICENSE='000000000000' THEN ELICENSE='            ';
 ELSE ELICENSE=RIGHT(LICENSE);
```

DATA ELEMENT: STATE LICENSED IN

In the July 1993 Registry file, there was only one observation with an invalid postal code.

Flag	Definition	Criteria
F	Feasible	valid postal code
M	Missing	all other values

FLOWCHART FOR STATE LICENSED IN (ESTLICNS) EDITS



Note:

STLICENS denotes state licensed in.

```
/*************************************************************/
/*                                                           */
/* -RESEARCHER EDITS: STATE LICENSED IN                      */
/*                                                           */
/*************************************************************/


IF STLICENS IN ('AL','AK','AZ','AR','CA','CO','CT','DE','DC','FL',
                'GA','HI','ID','IL','IN','IA','KS','KY','LA','ME',
                'MD','MA','MI','MN','MS','MO','MT','NE','NV','NH',
                'NJ','NM','NY','NC','ND','OH','OK','OR','PA','RI',
                'SC','SD','TN','TX','UT','VT','VA','WA','WV','WI',
                'WY','PR','GU','VI')
   THEN ESTLICNS=STLICENS;
```

DATA ELEMENT: SPECIALTY CODE

The primary specialty code is a required data element, but the secondary is not. Although feasible, use of codes equal to 49 (miscellaneous) or 99 (unknown) occurs too frequently.

Data Format:

Carrier Record: PIC 9(02)
Registry File: PIC X(02)

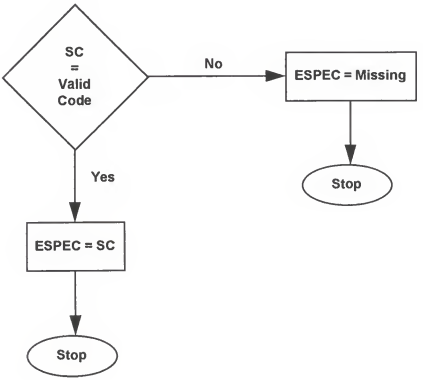
Flag	Definition	Criteria
F	Feasible	01-41,44,46,48,49,66,70,76-79, 81-86, 90-94,99
M	Missing	all other values

Notes:

(1) The value 00 is considered missing because of carrier instructions.

(2) In chapter 3 there are two sets of logical consistency comparisons between provider credentials and specialty code. One set is based on the above feasible values for specialty code. The other set includes recodes for osteopathic codes currently being phased out.

FLOWCHART FOR
PRIMARY SPECIALTY CODE (ESPEC) EDITS



Note:

SC denotes specialty code.

```
/*************************************************************/
/*                                                           */
/* -RESEARCHER EDITS: SPECIALTY CODES                        */
/*                                                           */
/*************************************************************/


../*   PRIMARY SPECIALTY CODE   */


 IF ('01'<=SPEC1<='41') OR (SPEC1 IN ('44','46','48','49','66','70','99'))
     OR ('76'<=SPEC1<='79') OR ('81'<=SPEC1<='86') OR ('90'<=SPEC1<='94')
     THEN ESPEC1=SPEC1;


../*   SECONDARY SPECIALTY CODE   */


 IF ('01'<=SPEC2<='41') OR (SPEC2 IN ('44','46','48','49','66','70','99'))
     OR ('76'<=SPEC2<='79') OR ('81'<=SPEC2<='86') OR ('90'<=SPEC2<='94')
     THEN ESPEC2=SPEC2;
```

DATA ELEMENT: BOARD CERTIFICATION

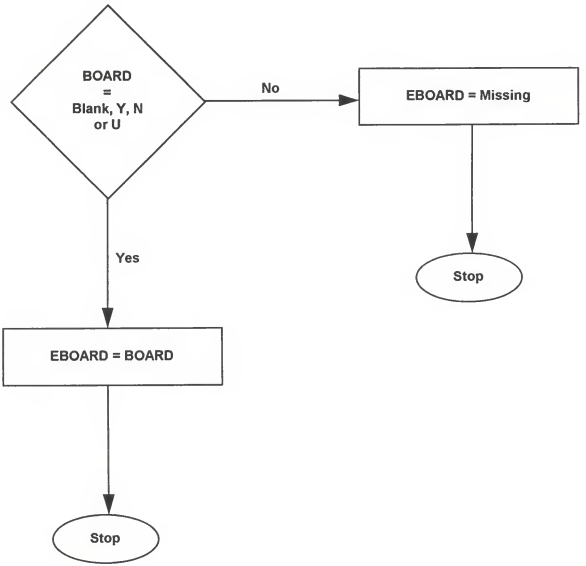
Neither the primary or secondary board certification variables are required data elements for the Registry.

Flag	Definition	Criteria
F	Feasible	blank, Y, N, or U
M	Missing	all other values

Note:

(1) Abbreviations: Y - Yes; N - No; U - Unknown.

FLOWCHART FOR BOARD CERTIFICATION (EBOARD) EDITS



```
 /*************************************************************/
 /*                                                           */
 /* -RESEARCHER EDITS: BOARD CERTIFICATION INDICATORS         */
 /*                                                           */
 /*************************************************************/

../*   PRIMARY SPECIALTY BOARD CERTIFICATION INDICATOR   */

 IF BOARD1 IN (' ','Y','N','U') THEN EBOARD1=BOARD1;

../*   SECONDARY SPECIALTY BOARD CERTIFICATION INDICATOR   */

 IF BOARD2 IN (' ','Y','N','U') THEN EBOARD2=BOARD2;
```

DATA ELEMENT: DEACTIVE, RESIDENT, INTERN OR PRACTICE (DRIP) STATUS

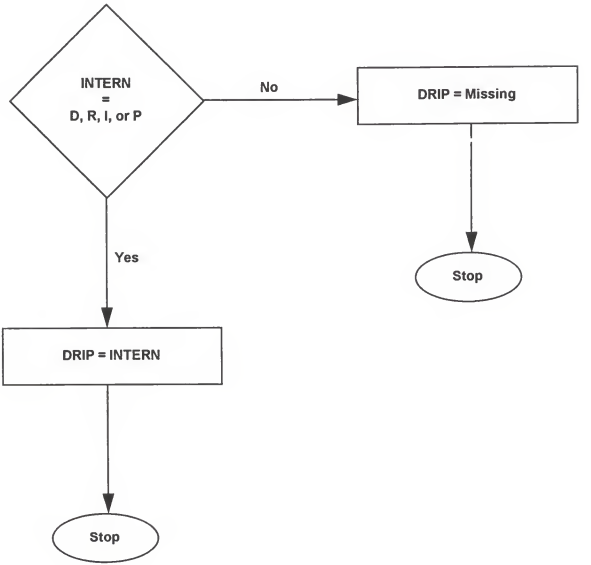
DRIP status is a required data element. In the July 1993 Registry file, 1,923 practice settings had blank or invalid values instead of one of the four valid values (*Carrier Analysis*, Exhibit 3-4).

Flag	Definition	Criteria
F	Feasible	D, R, I, or P
M	Missing	all other values

Note:

We used INTERN (TOLIC's name for the variable) as the name for the "input" variable but used the more familiar "DRIP" as the name for edited variable.

FLOWCHART FOR DRIP EDITS



Note:

INTERN denotes the DRIP status.

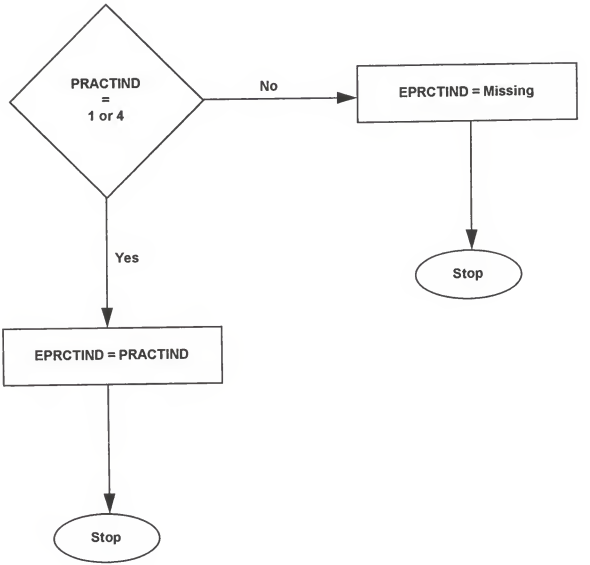
```
/*************************************************************/
/*                                                           */
/* -RESEARCHER EDITS: DRIP STATUS                            */
/*                                                           */
/*************************************************************/


IF INTERN IN ('D','R','I','P') THEN DRIP=INTERN;
```

DATA ELEMENT: GROUP PRACTICE INDICATOR

A few carriers still use the older, more comprehensive coding scheme for group practice indicator.

Flag	Definition	Criteria
F	Feasible	1 or 4
M	Missing	all other values

FLOWCHART FOR GROUP PRACTICE INDICATOR (EPRCTIND) EDITS



Note:

PRACTIND denotes the group practice indicator.

```
/*************************************************************/
/*                                                           */
/* -RESEARCHER EDITS: GROUP PRACTICE INDICATOR               */
/*                                                           */
/*************************************************************/


IF PRACTIND IN (1,4) THEN EPRCTIND=PRACTIND;
```

DATA ELEMENT: PHYSICIAN PARTICIPATION INDICATOR

The latest decision to accept Medicare beneficiaries on assignment is recorded.

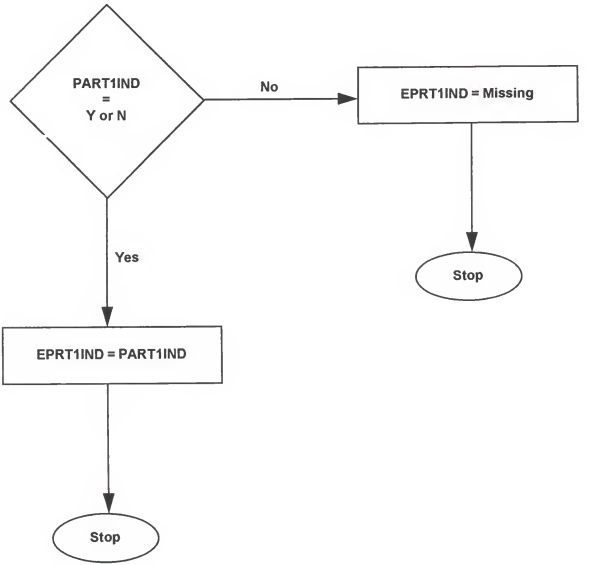
Flag assignment hierarchy, in descending order of priority:

Flag	Definition	Criteria
F	Feasible	Y or N
M	Missing	all other values

Note:

(1) Abbreviations: Y - Yes; N - No.

FLOWCHART FOR PHYSICIAN PARTICIPATION INDICATOR (EPRT1IND) EDITS



Note:

PART1IND denotes participation indicator.

```
/*************************************************************/
/*                                                           */
/* -RESEARCHER EDITS: PHYSICIAN PARTICIPATION INDICATOR      */
/*                                                           */
/*************************************************************/


IF PART1IND IN ('Y','N') THEN EPRT1IND=PART1IND;
```

3.0 EDITS TO OBTAIN LOGICAL CONSISTENCY BETWEEN VARIABLES

One of the potential problems that can occur in a provider file is logical inconsistency between data elements (variables) within a single record. An example of a logical inconsistency is when the provider status variable indicates that the provider is a physician or an osteopath while the provider credential variable indicates that the provider is an optometrist. In this particular example, both of the variables have feasible values. It is not immediately clear, however, which of the variables is improperly coded.

This chapter specifies the logically consistent values for pairs of data elements. Flowcharts and SAS programs implementing the specifications are also included. The specifications for each pair of data elements have several subsections. First, for each pair of data elements for which comparisons are to be made, an example of an inconsistency is given. Next, specifications for consistent values are provided. These specifications are based upon the data edits presented in the previous chapter. Therefore it is necessary to first implement the editing algorithms described in Chapter 2. In this chapter, the flowcharts do not include the data integrity checks, but the SAS programs do include them. (For example, at the beginning of the flowchart for the logical consistency between birth year and graduation year, there is a test of whether the respective flag values of these two data elements are both equal to F, thus assuming that the flag variables have been created. The SAS code on the following page, however, includes the creation of these two flag variables.)

The logical comparisons between pairs of data elements are not limited to the strictly feasible values as defined in Chapter 2. In particular, some of the feasible values were assigned flag values other than F (feasible) or M (missing). For instance, specialty code value 99 was assigned the flag value U (unknown). This was necessary because 99 is supposed to be used only for medical doctors and osteopaths. Rather than collectively referring to codes F and U as

feasible, to avoid confusion, they are collectively referred to as *acceptable*. Codes not designated as *acceptable* are designated as *unacceptable*.

For each pair of data elements, acceptable values are compared as indicated in the specifications. They are either consistent or inconsistent. If one of the data elements has an unacceptable value, then it is not compared to an acceptable value. The logical consistency flag (LCF) consists of four values:

LCF	Criteria
N	One or both data elements have *unacceptable* values
K	Consistent *acceptable* values
I	Inconsistent *acceptable* values
blank	Values not compared (only for school code compared to graduation year)

The flowcharts indicate how the flag variables for each pair of data element values should be coded as K, I, or N. (A blank flag value is only used for schools other than medical schools when comparing graduation date to the open/close dates of medical schools.)

When inconsistencies arise, school codes and specialty codes are forced to take values that depend on the value of provider credentials. This is done because provider credentials are less susceptible to error than the other two variables. When birth year and graduation year do not agree, the edited value of graduation year depends on birth year on the supposition that birth year is more likely to be accurate than graduation year.

The SAS program that performs the logical consistency edits is in Appendix C. The SAS program segments that follow each flowchart should not be performed in isolation from the full program. As a prelude to the logical consistency edits, the program merges (by UPIN) the base records and business setting records created by the programs in Appendices A and B. This step is necessary because such variables as specialty code can vary by business setting. Also, as noted in Chapter 1, this step is part of the process that creates an analytic file.

LOGICAL CONSISTENCY: PROVIDER CREDENTIALS and SCHOOL CODE

An example of an inconsistency is when provider credentials indicate a physician and the school type indicates that the provider graduated from a podiatric school.

Provider credentials (ECRED):

a. ECRED has seven acceptable values: MD, DO, DDM, DDS, CH, DPM, OD

b. ECRED flag (PCF) has one acceptable value: F

School code flag (SF) has one acceptable value: F

For school codes with SF = F, the school codes can be collapsed (MSCHTYPE) as indicated in Exhibit 3-1. Because of their values, osteopathic school codes should be recoded before the medical school codes. Similarly, depending on your computer system's collating sequence, podiatric school codes may need to be recoded before the medical school codes.

Logically consistent values of PC and school recode (MSCHTYPE) are found in Exhibit 3-2.

If an inconsistency is found, the inconsistency is eliminated by assigning a catch-all school code (based on the value of credential). See Exhibit 3-3 for recodes.

EXHIBIT 3-1

SCHOOL RECODES

MSCHTYPE	School Type	School Codes
1	Medical	< 10000
2	Dental	10000-29999
3	Podiatry	30000-39999, 000XA, 000XB, 000XC, 000XD, 000XE, 000XF, 000XG, 000XH, 000XI, 000XJ, 000XK
4	Optometrist	40000-49999
5	Chiropractic	50000-59999
6	Foreign	99999
7	Osteopathic	00576, 01175, 01675, 01875, 02275, 02576, 02878, 02879, 03375, 03575, 03875, 03979, 04177, 04878, 05575
8	bad code	any other code

Note: The ranges are used to indicate listed codes (Exhibit 2-1) within the range. Therefore, school code integrity checks need to be performed before these recodes are utilized.

EXHIBIT 3-2

LOGICALLY CONSISTENT VALUES OF PROVIDER CREDENTIALS AND SCHOOL RECODE (MSCHTYPE)

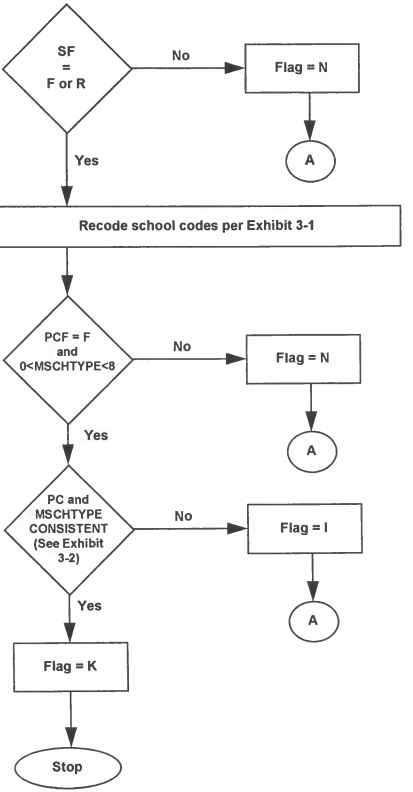
MSCHTYPE	ECRED
1	MD
2	DDM or DDS
3	DPM
4	OD
5	CH
6	MD, DDM, DDS, CH, or OD
7	DO

EXHIBIT 3-3

CREDENTIAL-BASED RECODES FOR SCHOOL CODE

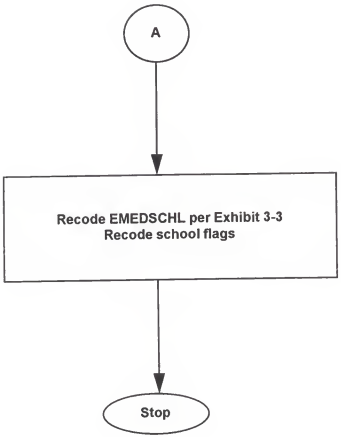
ECRED	SCHOOL CODE
MD	09098
DO	09099
DDM	10000
DDS	20000
DPM	39999
OD	49999
CH	50000
any other	missing

FLOWCHART FOR THE LOGICAL CONSISTENCY BETWEEN PROVIDER CREDENTIALS AND SCHOOL TYPE



Notes:

SF denotes school flag;
PC denotes provider credentials;
PCF denotes provider credentials flag.


A
Recode EMEDSCHL per Exhibit 3-3
Recode school flags
Stop

```
/*********************************************************************/
/                                                                     */
/  LOGICAL CONSISTENCY BETWEEN:                                       */
/                                                                     */
/* EDITED PROVIDER CREDENTIALS (ECRED) & EDITED SCHOOL CODE (EMEDSCHL) */
/                                                                     */
/*********************************************************************/

IF ECRED IN ('MD ','DO ','DDM','DDS','DPM','CH ','OD ') THEN HCREDNTL='F';
ELSE HCREDNTL='M';

IF EMEDSCHL IN ('00102','00104','00106','00301','00401','00402',
               '00501','00502','00504','00505','00506','00507',
               '00508','00509','00511','00512','00513','00514',
               '00515','00516','00517','00518','00519','00576',
               '00702','00705','00801','00802','01001','01002',
               '01003','01102','01103','01104','01175','01201',
               '01205','01209','01211','01212','01218','01219',
               '01221','01222','01401','01601','01602','01604',
               '01605','01606','01608','01609','01610','01611',
               '01613','01614','01615','01616','01617','01618',
               '01619','01622','01623','01642','01643','01644',
               '01645','01675','01705','01708','01709','01717',
               '01718','01720','01801','01803','01804','01806',
               '01808','01810','01875','01902','01903','01904',
               '01907','02001','02002','02004' '02005','02007',
               '02008','02009','02011','02012','02101','02104',
               '02105','02106','02201','02275','02301','02303',
               '02304','02305','02306','02307','02308','02309',
               '02311','02312','02401','02405','02406','02407',
               '02415','02416','02501','02505','02507','02508',
               '02512','02576','02604','02605','02607','02608',
               '02701','02702','02801','02802','02803','02805',
               '02807','02808','02810','02820','02822','02826',
               '02828','02830','02833','02834','02835','02843',
               '02844','02845','02846','02878','02879','03004',
               '03005','03006','03007','03101','03201','03305',
               '03306','03375','03401','03501','03503','03506',
               '03508','03509','03510','03511','03513','03515',
               '03519','03520','03543','03545','03546','03547',
               '03548','03575','03601','03603','03604','03605',
               '03607','03608','03701','03801','03802','03803',
               '03806','03808','03809','03811','03819','03823',
               '03825','03826','03840','03841','03843','03844',
               '03845','03875','03901','03905','03979','04001',
               '04002','04101','04102','04107','04109','04111',
               '04112','04113','04114','04177','04201','04202',
               '04203','04204','04301','04501','04504','04601',
               '04701','04705','04706','04707','04708','04709',
```

```
                '04710','04711','04713','04714','04715','04720',
                '04802','04803','04804','04805','04806','04807',
                '04812','04813','04814','04815','04816','04878',
                '04901','05002','05101','05104','05106','05107',
                '05404','05415','05501','05502','05575','05602',
                '05603','05605','05606','000XF','000XG',
                '000XH','000XI','30000','30100','30200','30300',
                '30400','30500','30600','40000','40010','40020',
                '40030','40040','40050','40060','40070','40080',
                '40090','40100','40110','40120','40130','40140',
                '40150','04160','40170','50000','50010','50020',
                '50030','50040','50050','50060','50065','50070',
                '50080','50090','50100','50110','50120','50130',
                '50140','50150','50160','50170','50180','50190',
                '50200','50210','50220','50230','50240','50250',
                '50260','50270','50280','50290','50300','50310',
                '99999','10000','20000'
                '000XA','000XB','000XC','000XD','000XE','000XJ','000XK')
                THEN HMEDSCHL='F';
 ELSE HMEDSCHL='M';

IF HMEDSCHL='F' THEN DO;
  IF EMEDSCHL IN ('00576','01175','01675','01875','02275','02576',
                  '02878','02879','03375','03575','03875','03979'
                  '04177','04878','05575') THEN MSCHTYPE=7;
   ELSE IF ('30000'<=EMEDSCHL<='39999') &
          (EMEDSCHL IN ('000XA','000XB','000XC','000XD','000XE',
                        '000XF','000XG','000XH','000XI','000XJ','000XK'))
                        THEN MSCHTYPE=3;
   ELSE IF EMEDSCHL<'10000' THEN MSCHTYPL=1;
   ELSE IF '10000'<=EMEDSCHL<='29999' THEN MSCHTYPE=2;
   ELSE IF '40000'<=EMEDSCHL<='49999' THEN MSCHTYPE=4;
   ELSE IF '50000'<=EMEDSCHL<='59999' THEN MSCHTYPE=5;
   ELSE IF EMEDSCHL='99999' THEN MSCHTYPE=6;
   ELSE MSCHTYPE=8;
END;

IF (MSCHTYPE=1 & ECRED='MD ') OR
   (MSCHTYPE=2 & (ECRED IN ('DDM','DDS'))) OR
   (MSCHTYPE=3 & ECRED='DPM') OR
   (MSCHTYPE=4 & ECRED='OD ') OR
   (MSCHTYPE=5 & ECRED='CH ') OR
   (MSCHTYPE=6 & (ECRED IN ('MD ','DDM','DDS','CH ','OD '))) OR
   (MSCHTYPE=7 & ECRED='DO ') THEN CSLCF='K';
 ELSE IF HMEDSCHL^='F' OR HCREDNTL^='F' THEN CSLCF='N';
 ELSE CSLCF='I';
```

```
/*****************************************/
/* ASSIGNMENT ALGORITHM (ECRED & EMEDSCHL) */
/*****************************************/

EHMDSCHL=HMEDSCHL;
EMSCHTYP=MSCHTYPE;

IF CSLCF IN ('I','N') THEN DO;
   IF ECRED='MD ' THEN DO;
      EMEDSCHL='09098';
      EHMDSCHL='F';
      EMSCHTYP=1;
   END;
    ELSE IF ECRED='DO ' THEN DO;
      EMEDSCHL='09099';
      EHMDSCHL='F';
      EMSCHTYP=7;
    END;
    ELSE IF ECRED='DDM' THEN DO;
      EMEDSCHL='10000';
      EHMDSCHL='F';
      EMSCHTYP=2;
    END;
    ELSE IF ECRED='DDS' THEN DO;
      EMEDSCHL='20000';
      EHMDSCHL='F';
      EMSCHTYP=2;
    END;
    ELSE IF ECRED='DPM' THEN DO;
      EMEDSCHL='39999';
      EHMDSCHL='F';
      EMSCHTYP=3;
    END;
    ELSE IF ECRED='OD ' THEN DO;
      EMEDSCHL='49999';
      EHMDSCHL='F';
      EMSCHTYP=4;
    END;
    ELSE IF ECRED='CH ' THEN DO;
      EMEDSCHL='50000';
      EHMDSCHL='F';
      EMSCHTYP=5;
    END;
END;
```

LOGICAL CONSISTENCY: BIRTH YEAR AND GRADUATION YEAR

An example of an inconsistency is when the graduation year and birth year imply a very young graduation age (e.g., 15 years old) or very old graduation age (e.g., 65 years old).

The birth year flag (BYF) has one acceptable value: F. The graduation year flag (GYF) has one acceptable value: F.

Graduation age (GRADAGE) is equal to graduation year minus birth year.

Birth year and graduation year are consistent when:

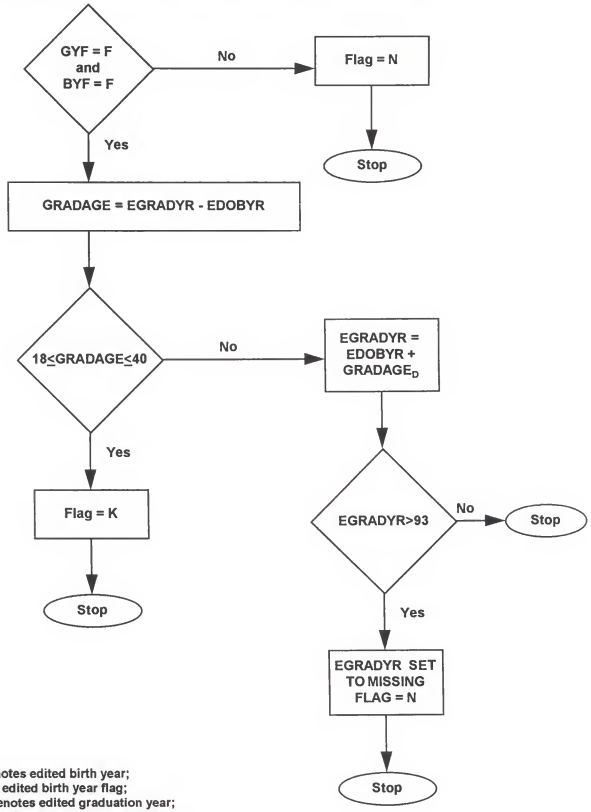
$$18 \leq \text{GRADAGE} \leq 40$$

Notes:

(1) The lower bound for an acceptable graduation age is less than implied by the birth year edits (Chapter 2). The lower bound of 18 was used because there may be a small number of providers that actually graduated between ages 18 and 23.

If an inconsistency is found, the edited value for graduation year is imputed by the sum of birth year and the graduation age assignments based on birth year (Chapter 2). If the imputed graduation year is greater than 93, then the edited graduation year is assigned a missing value code.

FLOWCHART FOR LOGICAL CONSISTENCY BETWEEN BIRTH YEAR AND GRADUATION YEAR



Notes:

EDOBYR denotes edited birth year;
BYF denotes edited birth year flag;
EGRADYR denotes edited graduation year;
GYF denotes edited graduation year flag;
GRADAGE denotes graduation age;
"D" subscript on GRADAGE denotes imputed value.

```
/*****************************************************/
/                                                     */
/  LOGICAL CONSISTENCY BETWEEN:                       */
/                                                     */
/* BIRTH YEAR (EDOBYR) AND GRADUATION YEAR (EGRADYR) */
/                                                     */
/*****************************************************/

IF ' 5'<=EDOBYR<='69' THEN BYF='F';
IF 30<=EGRADYR<=93 THEN GYF='F';

/*****************************************************/
/* GRADUATION AGE (GRADAGE) CALCULATION & IMPUTATION */
/*****************************************************/

IF BYF='F' & GYF='F' THEN DO;
   BGLCF='K';
   GRADAGE = EGRADYR - EDOBYR;
   IF GRADAGE<18 OR GRADAGE>40 THEN DO;
      IF ' 5'<=EDOBYR<='59' THEN GRADAGE=27;
      ELSE IF '60'<=EDOBYR<='65' THEN GRADAGE=26;
      ELSE IF '66'<=EDOBYR<='69' THEN GRADAGE=25;
      EGRADYR=EDOBYR + GRADAGE;
      IF EGRADYR>93 THEN DO;
         EGRADYR=.;
         IGRADYR=0;
         BGLCF='N';
      END;
   END;
END;
ELSE BGLCF='N';
```

LOGICAL CONSISTENCY: MEDICAL SCHOOL OPEN/CLOSE DATES and GRADUATION YEAR

We assume that program code has been created to identify medical schools (as distinct from other schools) where MSCHTYPE=1 denotes a medical school and that a graduation year flag (GYF) has been created (F is the acceptable value for GYF).

Because open and close dates are not available for defunct non-medical schools, such schools are excluded from comparisons and the comparison flag variable takes a *blank* value. The school code = '09098' denotes a "catch-all" medical school code. It is created when there is an inconsistency between provider's credentials and school code type – see the specifications on page 87 and Exhibit 3-3 on page 90.

Open and close dates for each school are in a separate file. (A flat file on diskette with school names and their codes and open/close dates has been created. A hardcopy list appears in Appendix F of this report.) Each open and close date includes the complete year value (i.e., 1856 instead of 56) to avoid century identification problems. Medical schools still in operation were given a pseudo closure date of 3000.

The carrier record graduation year has a YY format. Therefore, 1900 was added to its value in order to make it comparable to the open/closure dates. (This procedure would have to be modified after 1999.)

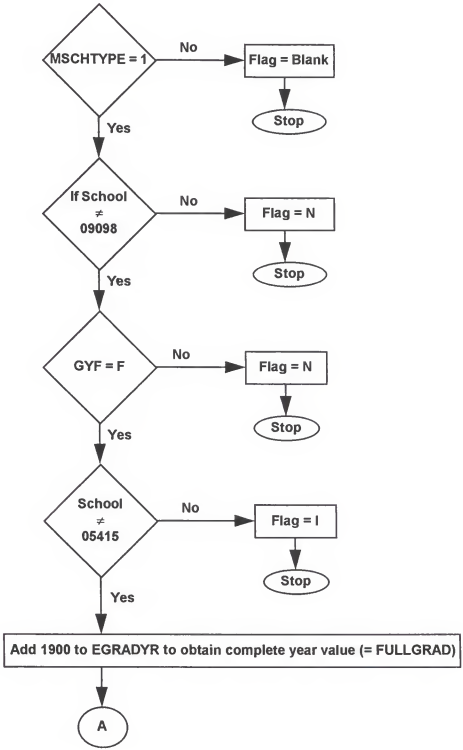
Open and/or closure dates could not be found for the Washington College of Physicians and Surgeons (code=05415) believed to have been located in Seattle, Washington. None of the contacted organizations and agencies ever heard of the school. Because open/closure dates are not available, the logical consistency flag will have a value of I (inconsistency between otherwise feasible values).

Logically consistent values for a given school code's open/close dates and graduation year:

$$\text{Open date} \leq \text{graduation year} \leq \text{Close date}$$

If an inconsistency is found, then EGRADYR is imputed if the adjustment to graduation year is small enough to prevent graduation age (GRADAGE) from being too high (above age 52) or too low (below age 18). If the implied graduation age is below 18 or above 52, then graduation year and graduation age are set to missing. We allowed the *implied* graduation age to exceed 40 only when the *actual* graduation age was missing or greater than 40. The maximum value 52 is implied by the constraints denoted in the flowchart and SAS program logic. The minimum age of 18 is explicitly indicated in the flowchart and SAS program.

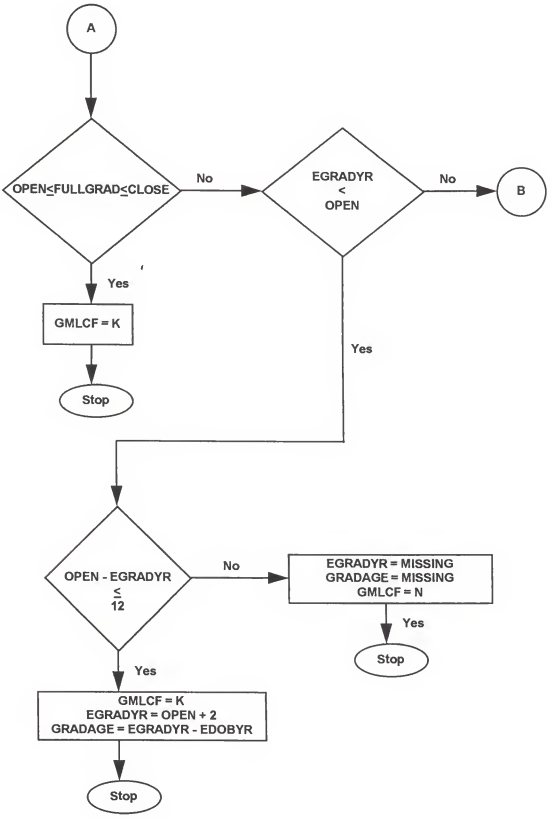
FLOWCHART FOR LOGICAL CONSISTENCY BETWEEN GRADUATION YEAR AND MEDICAL SCHOOL OPEN/CLOSE DATES



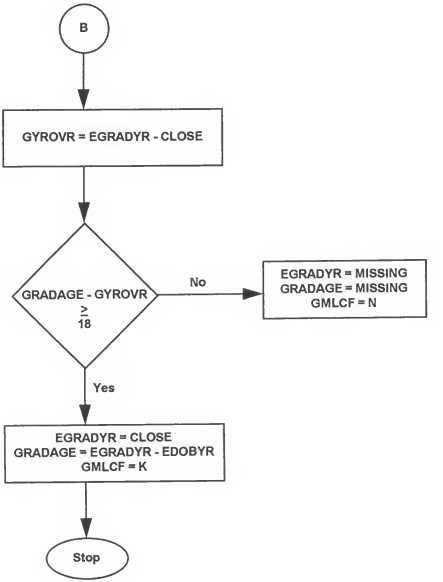
Notes:

GYF denotes edited graduation year flag;
SCHOOL denotes edited school code;
EGRADYR denotes edited graduation year;
OPEN denotes school open date;
CLOSE denotes school closure date.

FLOWCHART FOR LOGICAL CONSISTENCY BETWEEN GRADUATION YEAR AND MEDICAL SCHOOL OPEN/CLOSE DATES



FLOWCHART FOR LOGICAL CONSISTENCY BETWEEN GRADUATION YEAR AND MEDICAL SCHOOL OPEN/CLOSE DATES



```
/*************************************************************************/
/                                                                         */
/  LOGICAL CONSISTENCY BETWEEN:                                           */
/                                                                         */
/* GRADUATION YEAR (EGRADYR) & MED SCHOOL OPEN/CLOSE DATES (OPEN,CLOSE)  */
/* & MED SCHOOL TYPE (EMSCHTYP)                                          */
/                                                                         */
/*************************************************************************/

 IF EMSCHTYP^=1 THEN GMLCF=' ';
  ELSE IF EMEDSCHL='09098' OR BGLCF^='K' THEN GMLCF='N';
    ELSE IF EMEDSCHL='05415' THEN GMLCF='I';
     ELSE DO;
       FULLGRAD=1900+EGRADYR;
       IF OPENYR<=FULLGRAD<=CLOSEYR THEN GMLCF='K';
        ELSE IF FULLGRAD<OPENYR THEN DO;
         IF OPENYR - FULLGRAD <= 12 THEN DO;
          EGRADYR=(OPENYR-1900) + 2;
          GRADAGE=EGRADYR - EDOBYR;
          GMLCF='K';
         END;
         ELSE DO;
          EGRADYR=.;
          GRADAGE=.;
          GMLCF='N';
         END;
        END;
        ELSE DO;
         GYROVR = FULLGRAD - CLOSEYR;
         IF GRADAGE-GYROVR >= 18 THEN DO;
          EGRADYR = CLOSEYR - 1900;
          GRADAGE = EGRADYR - EDOBYR;
          GMLCF='K';
         END;
         ELSE DO;
          EGRADYR=.;
          GRADAGE=.;
          GMLCF='N';
         END;
        END;
   END;
END;
```

LOGICAL CONSISTENCY: PROVIDER CREDENTIALS and SPECIALTY CODE

An example of an inconsistency is when the specialty code indicates a physician specialty while the provider credential indicates a chiropractor.

Provider credentials (ECRED):

a. ECRED has 7 acceptable values: MD, DO, DDM, DDS, CH, DPM, OD

b. ECRED flag (PCF) has 1 acceptable value: F

Specialty code flag (SPF) has 3 acceptable values: F, S, or U (see Note 2 below).

Specialty code needs to be re-coded as SPECS (see Exhibit 3-4). Logically consistent values of PC and SPECS are shown in Exhibit 3-5.

Notes:

(1) A value of 10 was assigned for SPECS to catch any erroneous specialty code that was not picked by the variable integrity screens.

(2) The coding for the specialty code flag (SPF) is not included in the flowchart but is included in the accompanying SAS program.

(3) A second version of the edited specialty code (ESPEC1B) is created in the SAS program (but not in the flowchart). For ESPEC1B, the retired osteopathic codes were assigned their equivalent medical doctor code (per HCFA instructions to carriers in October 1991). The assignments are in Exhibit 3-6. (Note that all codes 17 were assigned to 04 instead of being split between 04 and 18.)

Between 1991 and 1992, out of 181 providers who left specialty code 17, 60 went to 04 (otolaryngology) and the rest went to 18 (ophthalmology).

(4) The data format for specialty code is different for the carrier record and the Registry file (see Chapter 1). Since the Registry version is alphanumeric, the accompanying SAS program includes some logic to recode blank fields as missing values and to recode the value 'b0' as zero (the lower case b indicates a blank).

If an inconsistency is found, the inconsistency is eliminated by assigning specialty codes (based on the value of credential). See Exhibit 3-7 for recodes (a code equal to 99 indicates an unknown specialty).

EXHIBIT 3-4

RECODE OF SPECIALTY CODE

SPECS	SPECIALTY CODE	PROVIDER TYPE
1	01, 02, 03, 04, 05, 06, 07, 08, 10, 11, 13, 14, 16, 18, 20, 22, 24, 25, 26, 28, 29, 30, 33, 34, 36, 37, 38, 39, 40, 44, 46, 66, 70, 76, 77, 78, 79, 81, 82, 83, 84, 86, 90, 91, 92, 93, 94	Medical Doctor
2	09, 12, 15, 17 ,21, 23, 27, 31, 32	Osteopath
3	19	Dentist
4	85	Dentist
5	35	Chiropractor
6	48	Podiatrist
7	41	Optometrist
8	49	Miscellaneous
9	99	Unknown
10	all other values	Non-UPIN Specialties and non-assigned specialty codes

EXHIBIT 3-5

LOGICALLY CONSISTENT VALUES OF PROVIDER CREDENTIALS (ECRED) and SPECS

ECRED	SPECS
MD	1, 8, or 9
DO	1, 2, 8, or 9
DDM or DDS	3 or 4
CH	5
DPM	6
OD	7

EXHIBIT 3-6

ASSIGNMENT OF OSTEOPATHIC SPECIALTY CODES FOR ESPEC1B

OLD	NEW
09	16
15	16
17	04
21	22
23	76
27	86
31	30
32	92

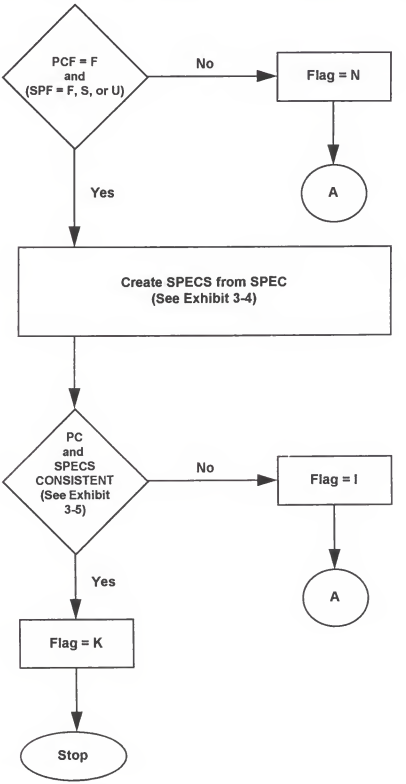
Note: All code 17s were assigned to 04 instead of being split between 04 and 18.

EXHIBIT 3-7

CREDENTIAL-BASED RECODES FOR SPECIALTY CODE

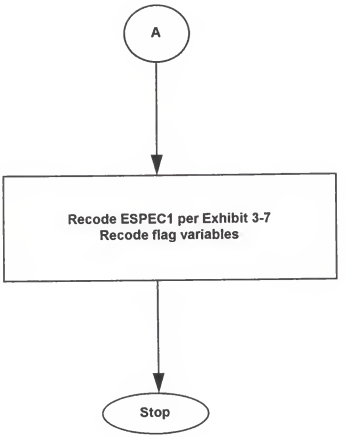
ECRED	EDITED SPECIALTY CODE
MD	99
DO	99
DDS	19
DDM	19
CH	35
DPM	48
OD	41

FLOWCHART FOR LOGICAL CONSISTENCY OF PROVIDER CREDENTIAL AND SPECIALTY CODE



Notes:

PC denotes provider credential;
PCF denotes provider credential flag;
SPEC denotes specialty code;
SPF denotes specialty code flag.


A
Recode ESPEC1 per Exhibit 3-7
Recode flag variables
Stop

```
/*************************************************************/
/                                                             */
/  LOGICAL CONSISTENCY BETWEEN:                               */
/                                                             */
/* PROVIDER CREDENTIALS (ECRED) & SPECIALTY CODE (ESPEC1)     */
/                                                             */
/*************************************************************/

IF ('01'<=ESPEC1<='41') OR (ESPEC1 IN ('44','46','48','66','70')) OR
   ('76'<=ESPEC1<='79') OR ('81'<=ESPEC1<='86') OR ('90'<=ESPEC1<='94')
   THEN HSPEC1='F';
ELSE IF ESPEC1 IN ('  ',' 0','00') THEN HSPEC1='M';
ELSE IF ESPEC1='99' THEN HSPEC1='U';
ELSE IF ESPEC1='49' THEN HSPEC1='S';
ELSE HSPEC1='E';

IF ESPEC1 IN
           ('01','02','03','04','05','06','07','08','10','11','13','14',
            '16','18','20','22','24','25','26','28','29','30','33',
            '34','36','37','38','39','40','44','46','66',
            '70','76','77','78','79','81','82','83','84','86','90',
            '91','92','93','94')
            THEN SPECS1=1;
 ELSE IF ESPEC1 IN
           ('09','12','15','17','21','23','27','31','32') THEN SPECS1=2;
 ELSE IF ESPEC1='19' THEN SPECS1=3;
 ELSE IF ESPEC1='85' THEN SPECS1=4;
 ELSE IF ESPEC1='35' THEN SPECS1=5;
 ELSE IF ESPEC1='48' THEN SPECS1=6;
 ELSE IF ESPEC1='41' THEN SPECS1=7;
 ELSE IF ESPEC1='49' THEN SPECS1=8;
 ELSE IF ESPEC1='99' THEN SPECS1=9;
 ELSE SPECS1=10;

IF (ECRED='MD ' & (SPECS1 IN (1,8,9))) OR
   (ECRED='DO ' & (SPECS1 IN (1,2,8,9))) OR
   ((ECRED IN ('DDM','DDS')) & (SPECS1 IN (3,4))) OR
   (ECRED='CH ' & SPECS1=5) OR
   (ECRED='DPM' & SPECS1=6) OR
   (ECRED='OD ' & SPECS1=7) THEN CPLCF='K';
 ELSE IF HCREDNTL^='F' OR (HSPEC1 NOT IN ('F','S','U')) THEN CPLCF='N';
 ELSE CPLCF='I';

/*****************************************/
/* ASSIGNMENT ALGORITHM (ECRED & EESPEC1) */
/*****************************************/

ESPECS1=SPECS1;
EHSPEC1=HSPEC1;
```

```
IF CPLCF IN ('I','N') THEN DO;
   IF ECRED IN ('MD ','DO ') THEN DO;
      ESPEC1='99';
      ESPECS1=9;
      EHSPEC1='U';
   END;
    ELSE IF ECRED IN ('DDM','DDS') THEN DO;
      ESPEC1='19';
      ESPECS1=3;
      EHSPEC1='F';
    END;
    ELSE IF ECRED='CH ' THEN DO;
      ESPEC1='35';
      ESPECS1=5;
      EHSPEC1='F';
    END;
    ELSE IF ECRED='DPM' THEN DO;
      ESPEC1='48';
      ESPECS1=6;
      EHSPEC1='F';
    END;
    ELSE IF ECRED='OD ' THEN DO;
      ESPEC1='41';
      ESPECS1=7;
      EHSPEC1='F';
    END;
END;

IF (ECRED='MD ' & (ESPECS1 IN (1,8,9))) OR
   (ECRED='DO ' & (ESPECS1 IN (1,2,8,9)), OR
   ((ECRED IN ('DDM','DDS')) & (ESPECS1 IN (3,4))) OR
   (ECRED='CH ' & ESPECS1=5) OR
   (ECRED='DPM' & ESPECS1=6) OR
   (ECRED='OD ' & ESPECS1=7) THEN ECPLCF='K';
 ELSE IF HCREDNTL^='F' OR (EHSPEC1 NOT IN ('F','S','U')) THEN ECPLCF='N';
 ELSE ECPLCF='I';

/*****************************************/
/* ASSIGNMENT ALGORITHM (ECRED & ESPEC1B)  */
/*****************************************/

ESPEC1B=ESPEC1;
ESPECS1B=ESPECS1;
EHSPEC1B=EHSPEC1;

IF ECRED='DO ' THEN DO;
   ESPECS1B=1;
   IF ESPEC1 IN ('09','15') THEN ESPEC1B=16;
    ELSE IF ESPEC1='17' THEN ESPEC1B='04';
    ELSE IF ESPEC1='21' THEN ESPEC1B='22';
```

```
    ELSE IF ESPEC1='23' THEN ESPEC1B='76';
    ELSE IF ESPEC1='27' THEN ESPEC1B='86';
    ELSE IF ESPEC1='31' THEN ESPEC1B='30';
    ELSE IF ESPEC1='32' THEN ESPEC1B='92';
END;

IF (ECRED='MD ' & (ESPECS1B IN (1,8,9))) OR
   (ECRED='DO ' & (ESPECS1B IN (1,2,8,9))) OR
   ((ECRED IN ('DDM','DDS')) & (ESPECS1B IN (3,4))) OR
   (ECRED='CH ' & ESPECS1B=5) OR
   (ECRED='DPM' & ESPECS1B=6) OR
   (ECRED='OD ' & ESPECS1B=7) THEN ECPLCFB='K';
 ELSE IF HCREDNTL^='F' OR (EHSPEC1B NOT IN ('F','S','U')) THEN ECPLCFB='N';
 ELSE ECPLCFB='I';
```

LOGICAL CONSISTENCY: SPECIALTY CODE and SCHOOL CODE

An example of an inconsistency is when the specialty code indicates a chiropractor but the school code indicates the provider graduated from a school of optometry.

Specialty code flag (SPF) has three acceptable values: F, S, or U. School code flag (SF) has one acceptable value: F.

Specialty code needs to be re-coded as SPECS (see Exhibit 3-4)

For school codes with SF = F, the school codes can be collapsed (MSCHTYPE) as indicated in Exhibit 3-1. Because of their values, osteopathic school codes should be recoded before the medical school codes. Similarly, depending on your system's collating sequence, podiatric school codes may need to be recoded before the medical school codes.

Logically consistent values of SPECS and school recode (MSCHTYPE) are shown in Exhibit 3-8.

Note:

(1) The recoding of school code and specialty code based on credential should automatically make school code consistent with specialty code.

(2) The coding for the specialty code flag (SPF) is not included in the flowchart but is included in the accompanying SAS program.

(3) The data format for specialty code is different for the carrier record and the Registry file (see Chapter 1). Since the Registry version is alphanumeric, the accompanying SAS program includes some logic to recode blank fields as missing values and to recode the value 'b0' as zero (the lower case b indicates a blank).

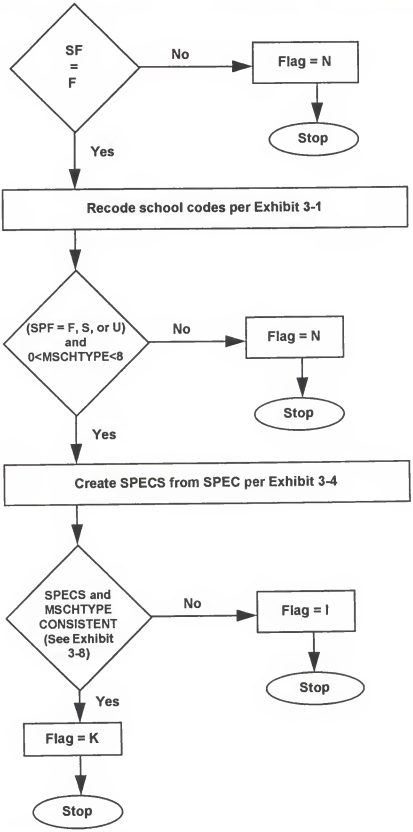
If an inconsistency is found, review the logical consistency programs for coding errors.

EXHIBIT 3-8

LOGICALLY CONSISTENT VALUES OF SPECIALTY RECODE (SPECS) AND SCHOOL RECODE (MSCHTYPE)

	MSCHTYPE	SPECS
Medical	1	1, 8, or 9
Dental	2	3 or 4
Podiatry	3	6
Optometrist	4	7
Chiropractic	5	5
Foreign	6	1, 3, 4, 5, 7, 8, or 9
Osteopathic	7	1, 2, 8, or 9

FLOWCHART FOR THE LOGICAL CONSISTENCY BETWEEN SPECIALTY CODE AND SCHOOL TYPE



Notes:

SF denotes school flag;
SPF denotes specialty code flag;
SPEC denotes specialty code.

```
/*************************************************************/
/                                                             */
/  LOGICAL CONSISTENCY BETWEEN:                               */
/                                                             */
/* SCHOOL TYPE (EMSCHTYP) AND PRIMARY SPECIALTY TYPE (ESPECS1) */
/                                                             */
/*************************************************************/

  IF (EMSCHTYP=1 & (ESPECS1 IN (1,8,9))) OR
     (EMSCHTYP=2 & (ESPECS1 IN (3,4))) OR
     (EMSCHTYP=3 & ESPECS1=6) OR
     (EMSCHTYP=4 & ESPECS1=7) OR
     (EMSCHTYP=5 & ESPECS1=5) OR
     (EMSCHTYP=6 & (ESPECS1 IN (1,3,4,5,7,8,9))) OR
     (EMSCHTYP=7 & (ESPECS1 IN (1,2,8,9))) THEN SPLCF='K';
   ELSE IF EHMDSCHL^='F' OR (EHSPEC1 NOT IN ('F','S','U'))
    THEN SPLCF='N';
   ELSE SPLCF='I';

RUN;
```

4.0 DETECTION OF MULTIPLE-UPIN PROVIDERS

4.1 Multiple-UPIN Providers

Each provider is supposed to have only one UPIN. However, many instances of providers with more than one UPIN have been found (*Carrier Analysis*, Chapter 8). Researchers may wish to drop or combine multiple-UPIN providers in their analyses. To assist researchers, this chapter presents several computerized algorithms by which to detect *cases* of multiple-UPIN providers. The computerized detection algorithms are in Appendix D. In this chapter, the word *cases* refers to sets of two or more UPINs identified by the detection algorithm. Cases identified by the detection algorithms, however, should be regarded as ***potential*** cases of multiple UPIN providers.

To assess whether an identified case might be a *false positive*, visual inspection of the data elements present in each record in each case is needed. It is recommended that the following data elements from the provider file be examined: state license number, state licensed in, credential, school code, graduation year, date of birth, the business address, the billing address, and all of the current name fields. In many instances, a discrepancy in one pair of *feasible* values (c.f., Chapter 2) will suffice to indicate that the potential multiple-UPIN case is a false positive. There are instances, however, in which some of the available information for a case indicates that it is a false positive while other information indicates that further investigation is warranted. (See the example presented in section 8.2.1. of the *Carrier Analysis*.) Depending on the variant of the screening variable used to detect multiple-UPIN providers, visual inspection of potential cases can be quite time consuming and expensive.

Since the computerized detection algorithms, by themselves, cannot distinguish between true positives and false positives, researchers will have to judge the consequences of: (1) dropping all suspect multiple-UPIN providers from their analyses, (2) combining the records of all suspect multiple-UPIN providers, and (3) ignoring multiple-UPIN providers (i.e.,

neither dropping nor combining their records). To assist researchers in making such judgments, four variants of the screening variable (algorithm) are presented.[1] The strengths and shortcomings of each of the components that are used in the four variants are described.[2] Also presented are summary statistics derived from tests of each algorithm. (Rather than use the edited variables created in Chapters 2 and 3, unedited variables from the original analytic file are used because the unedited values are the ones used by the *Registry* in assigning UPINs.)

4.2 Screening Variable Data Elements

To detect cases of multiple UPINs for a provider, a "screening" variable (ID_KEY) is created that is the result of concatenating parts of two or more data elements together to form one new variable. The screening variable, ID_KEY, always uses the date of birth and the first three letters of the last name of the provider. Date of birth is included because of the large number of individuals with common last names such as Jones and Smith. As a screening element, however, date of birth is only as good as its accuracy on the *Registry* file. Because of the inordinate number of providers with a date of birth of January 1, 1901, we believe that its usage is not valid. Additionally, although technically feasible, dates of birth prior to 1901 also seem implausible. Therefore, only *feasible* dates of birth after January 1, 1901 are used in the algorithm to detect multiple UPIN cases.[3] Because of concern about spelling variation (mistakes), only the first three letters of the last name are used (spaces, numbers, and special characters are not considered letters).

---

[1] A fifth algorithm is also described, but not necessarily recommended.

[2] See Chapter 8 in the *Carrier Analysis* for a more extensive discussion.

[3] Unlike the variable edits in Chapter 2, date of birth is allowed to have values prior to 1905.

Many different names begin with the same three letters, so that in some variants of the screening variable, LNAMESUM was used to avoid producing too many false positives. LNAMESUM is the sum of numbers assigned to the consonants in the last name. LNAMESUM was restricted to consonants because we believe that vowels are the most likely source of spelling variation.[4] (Therefore, LNAMESUM can fail to exclude some false positives, as when Ander*son* and Ander*sen* are tagged by the computer program as being identical.) In some circumstances LNAMESUM might overlook a good candidate for investigation. For instance, if an extra consonant was inadvertently added to a last name, as when an extra *L* is placed at the end of O'Neal (resulting in the spelling O'Neall instead of O'Neal) the screener values differ, and no multiple UPIN is detected.

The first initial of the first name (F1) was used in some variants of the screening variable because the number of providers with common last names is so high that even the inclusion of date of birth is not sufficient to distinguish between different providers. A difficulty with the first initial, of course, is due to the use of nicknames when the first letter of the nickname is not the same as the first letter of the proper first name (e.g., when *Bob* is used instead of *Robert*). All else being equal, the screener values would then differ, and no multiple UPIN would be detected.

4.3 Results of Implementing the Four Detection Algorithms

The SAS programs used to detect potential multiple-UPIN cases are in Appendix D. (The specifications for a generic program follow section 4.4). Each program produces a screening variable (ID_KEY_x) and a multiple-UPIN flag variable (MU_FL_x) where "x" denotes one of the variants of the screening variable. For each UPIN, the multiple-UPIN flag

[4] A value of one was assigned to B, two to C, three to D, and so forth, with 21 assigned to Z.

indicates: (1) no active practice settings for the UPIN and hence dropped from the detection routine, (2) date of birth infeasible or prior to January 2, 1901, (3) no other UPIN has the same value for ID_KEY_x, and (4) at least one other UPIN has the same value of ID_KEY_x. The frequencies for each of the multiple-UPIN flag values are shown in Table 4-1. The last row of the table indicates the total number of UPINs that are associated with potential multiple-UPIN cases. The number of potential cases is shown in the first column of Table 4-2.

The results of four variations on the detection algorithm produced a range of 3,530 to 25,308 potential multiple-UPIN cases (first column of Table 4-2). Depending on the presence of LNAMESUM, dropping the first initial of the first name from the screening variable identified anywhere from 4,700 to 20,000 additional potential cases. Similarly, omitting LNAMESUM resulted in 2,000 to 17,100 additional potential cases. Inclusion of LNAMESUM in the screening variable, in general, appears to result in the loss of too many good multiple-UPIN candidates.

To estimate the prevalence of false positives generated by each variant of the screening variable, as previously suggested, visual inspection was performed on the data elements of the first 50 potential cases produced by each of the four algorithms. The estimated rate of false positives ranged from 14 to 54 percent (second column of Table 4-2). As expected, the rate of false positives was highest when the screening variable had the fewest components. When the screening variable relied on the date of birth and the first three letters of the last name, the false positive rate was more than twice as high as the other variants of the detection algorithm. The number of potential cases shown in the first column of Table 4-2 was reduced by the estimated number of false positives, to produce the third column of Table 4-2, which shows the range of the *net* potential number of cases: 3,036 to 11,642 cases.

An examination of spelling variation was performed in conjunction with the identification of seemingly false positive cases. In the first instance, the spellings of the provider's last name were checked for minor discrepancies. For pairs of common last names

TABLE 4-1

FREQUENCIES OF SCREENING FLAG BY SCREENING VARIANT

SCREENING FLAG		SCREENING VARIABLE VARIANT				
Value	Definition	DOB, L1_3,F1, LNAMESUM	DOB, L1_3, LNAMESUM	DOB, L1_3, F1	DOB, L1_3	L1_3
1	No active practice settings for UPIN or DOB = 0	86,032	86,032	86,032	86,032	n/a
2	Birth date infeasible or prior to Jan. 2, 1901	20,018	20,018	20,018	20,018	n/a
3	No other UPIN has same value of ID_KEY	580,209	570,627	576,135	534,587	741
4	At least one other UPIN has same value of ID_KEY	7,096	16,678	11,170	52,718	19,277

Notes: 1. n/a indicates not applicable.
2. The screening variant in the last column was used only on observations with birth dates prior to Jan. 2, 1901 or infeasible dates.

Abbreviations:

C.A.I. - Computer Algorithm Identified
DOB - Date of Birth
L1_3 - First 3 letters of last name
F1 - First letter of first name
LNAMESUM - Sum of the numbers assigned to the consonants in the last name

TABLE 4-2

NUMBER OF POSSIBLE CASES WITH MULTIPLE UPINS

Screening Variable Elements*	C.A.I. Potential Number of Cases	Estimated Percent of False Positives	Net Potential Number of Cases
DOB,L1_3,F1,LNAMESUM	3,530	14 %	3,036
DOB,L1_3,LNAMESUM	8,201	26	6,069
DOB,L1_3,F1	5,540	20	4,432
DOB,L1_3	25,308	54	11,642

*See Table 4-1 for definitions of the screening variable elements.

Abbreviation: C.A.I. - Computer Algorithm Identified.

SOURCE: Physician Registry, July 1993 (Carrier Analysis, Table 8-1).

(e.g., Anderson and Andersen), unless other information pointed to a multiple-UPIN case, the case was marked as a false positive. We usually found, in actual cases of providers with more than one UPIN, that the provider's last name is not common. The absence or presence of an extra consonant is the apparent cause for the issuance of more than one UPIN to a provider in these cases. Therefore, such cases would not be marked as *seemingly* false positives. This examination helped produce an alternate assessment of the usefulness of the optional screening variable components. The two most important conclusions we drew from the examination were, once the computerized detection algorithms produced potential cases: (1) the results need to be visually inspected and (2) it is difficult, given the nature of the data, to incorporate other computerized data checks to reduce the number of false positives.

Selected characteristics of multiple-UPIN cases are presented in Table 4-3. Most of the potential cases involved multiple carriers. (Note, however, that since only the first active practice setting was used for a given UPIN, the term *solo* does not imply that there were absolutely no other practice settings submitted by other carriers.) While most of the cases of potential multiple-UPIN providers involved physicians and osteopaths, other providers were disproportionately represented in the potential cases.[5] That is, other providers account for about 11 percent of practice settings but represented from 13 to 36 percent of the UPINs in Table 4-3. The number of UPINs involved in the potential cases range from two to six, with the vast majority having two UPINs. It seems likely that false positives would account for most cases having three UPINs and all of the cases having four or more UPINs. The two Railroad Retirement Board (RRB) carriers often encounter significant difficulties in obtaining information from providers. Thus, it seemed likely that the RRB carriers would be present in many multiple-UPIN cases. As can be seen in the last two rows of Table 4-3, this expectation was borne out when only the multi-*carrier* cases were considered.

[5]Because of inconsistencies in the value of physician status among practice settings (*Carrier Analysis*, Table 6-1), the total number of UPINs involved in the multiple-UPIN cases, instead of the number of cases themselves, are presented in Table 4-3 for physician status.

TABLE 4-3

CHARACTERISTICS OF POSSIBLE MULTIPLE UPIN CASES

	SCREENING VARIABLE ELEMENTS			
	DOB, L1_3,F1 LNAMESUM	DOB,L1_3, LNAMESUM	DOB, L1_3,F1	DOB, L1_3
Total Number of Multiple UPIN Cases	3,530	8,201	5,540	25,308
Multiple UPIN Cases by Presence of Multiple Carriers				
Solo Carrier	845	1,503	1,067	2,192
Multiple Carriers	2,685	6,698	4,473	23,116
Physician Status (over all UPINs)				
MD, DO	4,906	11,807	7,538	39,386
All Others	2,190	4,871	3,632	13,332
Number of UPINs per Multiple-UPIN Case				
2 UPINs	3,494	7,947	5,451	23,446
3 UPINs	36	234	88	1,650
4 UPINs	0	18	1	185
5 UPINs	0	2	0	26
6 UPINs	0	0	0	1
Number of Multi-Carrier Cases in which RRB Carriers Present				
10072 - RRB Georgia	941	1,190	1,167	1,872
10074 - RRB Utah	768	1,134	949	1,816

Notes: 1. See Table 4-1 for definitions of the screening variable elements.
2. Because of inconsistencies in the value of physician status between practice settings (Carrier Analysis, Table 6-1), the total number of UPINs involved in the multiple-UPIN cases, instead of the number of cases themselves, are presented in this table for physician status.

Source: Physician Registry, July 1993 (Carrier Analysis, Table 8-2).

4.4 Provider Records Without Feasible Dates of Birth

For providers without feasible dates of birth (or prior to January 2, 1901), just use the first three letters of the provider's last name as the value of screening variable and modify the detection algorithm as necessary to exclude feasible dates (this is the aforementioned fifth variant of the screening variable). Of the 20,018 UPINs, 96.3 percent have a value of the screening variable that is equal to the value of the screening variable for at least one other UPIN. By comparison to the other variants of the screening variable, this percentage is quite high. Unlike like the other variants of the screening variable, however, inspection of data dumps was not performed in order to obtain estimates of the number of false positives. (Statistics for this fifth screening variable variant are not in Table 4-3.)

LOGIC TO DETECT MULTIPLE-UPIN PROVIDERS

Exclude from the detection run: (1) providers with no active practice settings, and (2) dates of birth prior to January 2, 1901 and all infeasible dates.

Create a screening variable by concatenating the date of birth, the first three letters of the provider's last name (use only letters) and, if wanted, the provider's first initial and/or LNAMESUM.

Sort the file by the screening variable.

If there is only one provider (observation) for a given value of the screening variable, then drop the observation from subsequent processing. The remaining observations constitute the pool of *potential* cases of multiple-UPIN providers. (Dividing the total number of remaining observations by two will provide an approximate number of potential multiple-UPIN cases.)

For each value of the screening variable, print information for each provider that could be useful for distinguishing individual providers from one another. We recommend, at a minimum, the following data elements to help determine whether the detected case should be further investigated as a potential multiple-UPIN provider:

- date of birth
- provider's last name
- provider's first name
- provider's middle initial
- provider's name suffix (e.g., Jr.)
- provider credentials
- physician status
- state licensed in
- state license number
- school code
- graduation year

APPENDIX A

SAS PROGRAM TO CREATE BASE RECORDS

```
  /*********************************************************/
  /*                                                       */
  /* -RESEARCHER EDITS: INDIVIDUAL VARIABLES               */
../*                     FOR BASE RECORD                   */
  /*                                                       */
  /*********************************************************/

 /* DATA IS SORTED BY UNIQUE PHYSICIAN IDENTIFIER (UPIN),*/
 /* RAILROAD BOARD INDICATOR FLAG (RRB), AND DESCENDING  */
 /* ORDER OF DATE SETTING ADDED (SETDATE).               */

DATA A; SET IN.JUL0293
       (KEEP=UPIN DOB DOD MEDSCHL GRADYR SANC1 SANC1DAT SANC1YRS
             CREDENTL PHYSSTAT LNAME FNAME MNAME SUFFIX
             SETDATE SETMDSCH SETDOB SETGRDYR CARRIER);

 /*******************************************/
 /* BOGUS OBSERVATION REMOVED BY CODE BELOW */
 /*******************************************/

 IF SETDATE>0 THEN SETYR = SUBSTR(SETDATE,7,2);

 IF SETYR='89' THEN SETYRADD=1;
 ELSE IF SETYR='90' ] SETYR='91' THEN SETYRADD=2;
 ELSE IF SETYR='92' ] SETYR='93' THEN SETYRADD=3;
 ELSE DELETE;

RRB=(CARRIER=10072 OR CARRIER=10074);

PROC SORT DATA=A;
 BY UPIN RRB DESCENDING SETDATE;
RUN;

DATA B(KEEP=UPIN SETDATE ESANC1 ESANCDAT ESANCYRS EMEDSCHL ECRED EPHYSTAT
           ELNAME EFNAME EMNAME ESUFFIX EDOB EDOBYR EDOD EDODYR EGRADYR);
     SET A;
     BY UPIN;
```

```
/*************************************************************************/
/* FOR EACH OF THE BASE RECORD VARIABLES, VALUES ARE ASSIGNED IN         */
/* THE ORDER OF HEADER, MOST RECENT PRACTICE SETTING, NEXT MOST          */
/* RECENT PRACTICE SETTING, ETC.. IN THE IF STATEMENTS BELOW, "E"        */
/* VARIABLES WHICH DO NOT NEED FEASIBILITY CHECKS ARE ASSIGNED           */
/* (NAME VARIABLES, SANCTION VARIABLES), AND "E" VARIABLES               */
/* THAT DO NEED FEASIBILITY CHECKS ARE INITIALIZED (CREDENTIAL,          */
/* PHYSICIAN STATUS, DATE OF BIRTH VARIABLES, DATE OF DEATH VARIABLES,   */
/* MEDICAL SCHOOL, AND GRADUATION YEAR) THEN RUN THROUGH THEIR RESPEC-   */
/* TIVE TESTS. NOTE THAT IF THE HEADER VERSION OF THE VARIABLE IS NOT    */
/* FEASIBLE, THE ACCOMPANYING SETTING VERSION IS TESTED FOR FEASIBILITY */
/* IMMEDIATELY. FEASIBLE RRB INFORMATION (CARRIER=10072 OR 10074) IS     */
/* USED ONLY IN CASES WHICH THE PROVIDER ONLY HAS PRACTICE SETTINGS      */
/* SUBMITTED BY THE RRB, OR WHEN FEASIBLE VALUES CANNOT BE OBTAINED      */
/* FROM OTHER CARRIERS.                                                  */
/*************************************************************************/


IF FIRST.UPIN THEN DO;
    ELNAME=LEFT(LNAME);
    EFNAME=LEFT(FNAME);
    EMNAME=LEFT(MNAME);
    ESUFFIX=LEFT(SUFFIX);
    ESANC1=SANC1;
    ESANCDAT=SANC1DAT;
    ESANCYRS=SANC1YRS;
    ECRED='   ';
    EPHYSTAT=.;
    EDOB=.;
    EDOD=.;
    EDOBYR='  ';
    EDODYR='  ';
    EMEDSCHL='     ';
    EGRADYR=.;
    LRRB=.;

 IF MEDSCHL IN ('00102','00104','00106','00301','00401','00402',
               '00501','00502','00504','00505','00506','00507',
               '00508','00509','00511','00512','00513','00514',
               '00515','00516','00517','00518','00519','00576',
               '00702','00705','00801','00802','01001','01002',
               '01003','01102','01103','01104','01175','01201',
               '01205','01209','01211','01212','01218','01219',
               '01221','01222','01401','01601','01602','01604',
               '01605','01606','01608','01609','01610','01611',
               '01613','01614','01615','01616','01617','01618',
               '01619','01622','01623','01642','01643','01644',
               '01645','01675','01705','01708','01709','01717',
               '01718','01720','01801','01803','01804','01806',
```

```
                '01808','01810','01875','01902','01903','01904',
                '01907','02001','02002','02004','02005','02007',
                '02008','02009','02011','02012','02101','02104',
                '02105','02106','02201','02275','02301','02303',
                '02304','02305','02306','02307','02308','02309',
                '02311','02312','02401','02405','02406','02407',
                '02415','02416','02501','02505','02507','02508',
                '02512','02576','02604','02605','02607','02608',
                '02701','02702','02801','02802','02803','02805',
                '02807','02808','02810','02820','02822','02826',
                '02828','02830','02833','02834','02835','02843',
                '02844','02845','02846','02878','02879','03004',
                '03005','03006','03007','03101','03201','03305',
                '03306','03375','03401','03501','03503','03506',
                '03508','03509','03510','03511','03513','03515',
                '03519','03520','03543','03545','03546','03547',
                '03548','03575','03601','03603','03604','03605',
                '03607','03608','03701','03801','03802','03803',
                '03806','03808','03809','03811','03819','03823',
                '03825','03826','03840','03841','03843','03844',
                '03845','03875','03901','03905','03979','04001',
                '04002','04101','04102','04107','04109','04111',
                '04112','04113','04114','04177','04201','04202',
                '04203','04204','04301','04501','04504','04601',
                '04701','04705','04706','04707','04708','04709',
                '04710','04711','04713','04714','04715','04720',
                '04802','04803','04804','04805','04806','04807',
                '04812','04813','04814','04815','04816','04878',
                '04901','05002','05101','05104' '05106','05107',
                '05404','05415','05501','05502','05575','05602',
                '05603','05605','05606','000XF','000XG',
                '000XH','000XI','30000','30100','30200','30300',
                '30400','30500','30600','40000','40010','40020',
                '40030','40040','40050','40060','40070','40080',
                '40090','40100','40110','40120','40130','40140',
                '40150','04160','40170','50000','50010','50020',
                '50030','50040','50050','50060','50065','50070',
                '50080','50090','50100','50110','50120','50130',
                '50140','50150','50160','50170','50180','50190',
                '50200','50210','50220','50230','50240','50250',
                '50260','50270','50280','50290','50300','50310',
                '99999','10000','20000',
                '000XA','000XB','000XC','000XD','000XE',
                '000XJ','000XK') THEN EMEDSCHL=MEDSCHL;

ELSE IF SETMDSCH IN ('00102','00104','00106','00301','00401','00402',
                '00501','00502','00504','00505','00506','00507',
                '00508','00509','00511','00512','00513','00514',
                '00515','00516','00517','00518','00519','00576',
                '00702','00705','00801','00802','01001','01002',
```

```
'01003','01102','01103','01104','01175','01201',
'01205','01209','01211','01212','01218','01219',
'01221','01222','01401','01601','01602','01604',
'01605','01606','01608','01609','01610','01611',
'01613','01614','01615','01616','01617','01618',
'01619','01622','01623','01642','01643','01644',
'01645','01675','01705','01708','01709','01717',
'01718','01720','01801','01803','01804','01806',
'01808','01810','01875','01902','01903','01904',
'01907','02001','02002','02004','02005','02007',
'02008','02009','02011','02012','02101','02104',
'02105','02106','02201','02275','02301','02303',
'02304','02305','02306','02307','02308','02309',
'02311','02312','02401','02405','02406','02407',
'02415','02416','02501','02505','02507','02508',
'02512','02576','02604','02605','02607','02608',
'02701','02702','02801','02802','02803','02805',
'02807','02808','02810','02820','02822','02826',
'02828','02830','02833','02834','02835','02843',
'02844','02845','02846','02878','02879','03004',
'03005','03006','03007','03101','03201','03305',
'03306','03375','03401','03501','03503','03506',
'03508','03509','03510','03511','03513','03515',
'03519','03520','03543','03545','03546','03547',
'03548','03575','03601','03603','03604','03605',
'03607','03608','03701','03801','03802','03803',
'03806','03808','03809','03811','03819','03823',
'03825','03826','03840','03841','03843','03844',
'03845','03875','03901','03905','03979','04001',
'04002','04101','04102','04107','04109','04111',
'04112','04113','04114','04177','04201','04202',
'04203','04204','04301','04501','04504','04601',
'04701','04705','04706','04707','04708','04709',
'04710','04711','04713','04714','04715','04720',
'04802','04803','04804','04805','04806','04807',
'04812','04813','04814','04815','04816','04878',
'04901','05002','05101','05104','05106','05107',
'05404','05415','05501','05502','05575','05602',
'05603','05605','05606','000XF','000XG',
'000XH','000XI','30000','30100','30200','30300',
'30400','30500','30600','40000','40010','40020',
'40030','40040','40050','40060','40070','40080',
'40090','40100','40110','40120','40130','40140',
'40150','04160','40170','50000','50010','50020',
'50030','50040','50050','50060','50065','50070',
'50080','50090','50100','50110','50120','50130',
'50140','50150','50160','50170','50180','50190',
'50200','50210','50220','50230','50240','50250',
'50260','50270','50280','50290','50300','50310',
'99999','10000','20000',
```

```
            '000XA','000XB','000XC','000XD','000XE',
            '000XJ','000XK') THEN EMEDSCHL=SETMDSCH;

CRED=LEFT(CREDENTL);
IF CRED IN ('MD ','DO ','DDS','DDM','DPM','CH ','OD ') THEN ECRED=CRED;

IF ECRED IN ('MD ','DO ') THEN EPHYSTAT=1;
 ELSE IF ECRED IN ('DDS','DDM','DPM','CH ','OD ') THEN EPHYSTAT=2;

DODYR = SUBSTR(DOD,7,2);
DODMMDD = SUBSTR(DOD,9,4);

IF DOD=999999 OR DOD=0 THEN DO;
 EDODYR=' 0';
 EDOD=0;
END;
 ELSE IF DODYR IN ('80','81','82','83','84','85','86','87'
                   '88','89','90','91','92','93') THEN DO;
   EDODYR=DODYR;
   IF ('80'<=DODYR<='92' &
    (('0101'<=DODMMDD<='0131') ] ('0201'<=DODMMDD<='0228') ]
    (DODMMDD='0229' & (DODYR/4=INT(DODYR/4))) ]
    ('0301'<=DODMMDD<='0331') ] ('0401'<=DODMMDD<='0430') ]
    ('0501'<=DODMMDD<='0531') ] ('0601'<=DODMMDD<='0630') ]
    ('0701'<=DODMMDD<='0731') ] ('0801'<=DODMMDD<='0831') ]
    ('0901'<=DODMMDD<='0930') ] ('1001'<=DODMMDD<='1031') ]
    ('1101'<=DODMMDD<='1130') ] ('1201'<=DODMMDD<='1231'))) ]
    (DODYR='93' &
    (('0101'<=DODMMDD<='0131') ] ('0201 <=DODMMDD<='0229') ]
    ('0301'<=DODMMDD<='0331') ] ('0401'<=DODMMDD<='0430') ]
    ('0501'<=DODMMDD<='0531') ] ('0601'<=DODMMDD<='0630')))
    THEN EDOD=DOD;
  END;

  DOBYR = SUBSTR(DOB,7,2);
  DOBMMDD = SUBSTR(DOB,9,4);
  SETDOBYR = SUBSTR(SETDOB,7,2);
  SDOBMMDD = SUBSTR(SETDOB,9,4);

  IF ' 5'<=DOBYR<='69' THEN DO;
    EDOBYR=DOBYR;
    IF ('0101'<=DOBMMDD<='0131') ] ('0201'<=DOBMMDD<='0228') ]
       (DOBMMDD='0229' & (DOBYR/4=INT(DOBYR/4))) ]
       ('0301'<=DOBMMDD<='0331') ] ('0401'<=DOBMMDD<='0430') ]
       ('0501'<=DOBMMDD<='0531') ] ('0601'<=DOBMMDD<='0630') ]
       ('0701'<=DOBMMDD<='0731') ] ('0801'<=DOBMMDD<='0831') ]
       ('0901'<=DOBMMDD<='0930') ] ('1001'<=DOBMMDD<='1031') ]
       ('1101'<=DOBMMDD<='1130') ] ('1201'<=DOBMMDD<='1231')
       THEN EDOB=DOB;
  END;
```

```
    IF EDOB=. THEN DO;
     IF ' 5'<=SETDOBYR<='69' THEN DO;
      EDOBYR=SETDOBYR;
      IF ('0101'<=SDOBMMDD<='0131') ] ('0201'<=SDOBMMDD<='0228') ]
         (SDOBMMDD='0229' & (SETDOBYR/4=INT(SETDOBYR/4))) ]
         ('0301'<=SDOBMMDD<='0331') ] ('0401'<=SDOBMMDD<='0430') ]
         ('0501'<=SDOBMMDD<='0531') ] ('0601'<=SDOBMMDD<='0630') ]
         ('0701'<=SDOBMMDD<='0731') ] ('0801'<=SDOBMMDD<='0831') ]
         ('0901'<=SDOBMMDD<='0930') ] ('1001'<=SDOBMMDD<='1031') ]
         ('1101'<=SDOBMMDD<='1130') ] ('1201'<=SDOBMMDD<='1231')
         THEN EDOB=SETDOB;
     END;
    END;

    IF 30<=GRADYR<=93 THEN EGRADYR=GRADYR;
    ELSE IF 30<=SETGRDYR<=93 THEN EGRADYR=SETGRDYR;

END;

 /*****************************************************************/
 /* THOSE UPIN WHOSE HEADER AND MOST RECENT PRACTICE SETTING      */
 /* FAIL FEASIBILITY TESTS ARE PASSED THROUGH THE NEXT DO LOOP.   */
 /*****************************************************************/

ELSE DO;
  IF EMEDSCHL='     ' THEN DO;
   IF SETMDSCH IN ('00102','00104','00106','00301','00401','00402',
                   '00501','00502','00504','00505','00506','00507',
                   '00508','00509','00511','00512','00513','00514',
                   '00515','00516','00517','00518','00519','00576',
                   '00702','00705','00801','00802','01001','01002',
                   '01003','01102','01103','01104','01175','01201',
                   '01205','01209','01211','01212','01218','01219',
                   '01221','01222','01401','01601','01602','01604',
                   '01605','01606','01608','01609','01610','01611',
                   '01613','01614','01615','01616','01617','01618',
                   '01619','01622','01623','01642','01643','01644',
                   '01645','01675','01705','01708','01709','01717',
                   '01718','01720','01801','01803','01804','01806',
                   '01808','01810','01875','01902','01903','01904',
                   '01907','02001','02002','02004','02005','02007',
                   '02008','02009','02011','02012','02101','02104',
                   '02105','02106','02201','02275','02301','02303',
                   '02304','02305','02306','02307','02308','02309',
                   '02311','02312','02401','02405','02406','02407',
                   '02415','02416','02501','02505','02507','02508',
                   '02512','02576','02604','02605','02607','02608',
                   '02701','02702','02801','02802','02803','02805',
                   '02807','02808','02810','02820','02822','02826',
```

```
              '02828','02830','02833','02834','02835','02843',
              '02844','02845','02846','02878','02879','03004',
              '03005','03006','03007','03101','03201','03305',
              '03306','03375','03401','03501','03503','03506',
              '03508','03509','03510','03511','03513','03515',
              '03519','03520','03543','03545','03546','03547',
              '03548','03575','03601','03603','03604','03605',
              '03607','03608','03701','03801','03802','03803',
              '03806','03808','03809','03811','03819','03823',
              '03825','03826','03840','03841','03843','03844',
              '03845','03875','03901','03905','03979','04001',
              '04002','04101','04102','04107','04109','04111',
              '04112','04113','04114','04177','04201','04202',
              '04203','04204','04301','04501','04504','04601',
              '04701','04705','04706','04707','04708','04709',
              '04710','04711','04713','04714','04715','04720',
              '04802','04803','04804','04805','04806','04807',
              '04812','04813','04814','04815','04816','04878',
              '04901','05002','05101','05104','05106','05107',
              '05404','05415','05501','05502','05575','05602',
              '05603','05605','05606','000XF','000XG',
              '000XH','000XI','30000','30100','30200','30300',
              '30400','30500','30600','40000','40010','40020',
              '40030','40040','40050','40060','40070','40080',
              '40090','40100','40110','40120','40130','40140',
              '40150','04160','40170','50000','50010','50020',
              '50030','50040','50050','50060','50065','50070',
              '50080','50090','50100','50110','50120','50130',
              '50140','50150','50160','50170','50180','50190',
              '50200','50210','5022C','50230','50240','50250',
              '50260','50270','50280','50290','50300','50310',
              '99999','10000','20000',
              '000XA','000XB','000XC','000XD','000XE',
              '000XJ','000XK') THEN EMEDSCHL=SETMDSCH;
END;

IF ECRED='    ' THEN DO;
 CRED=LEFT(CREDENTL);
 IF CRED IN ('MD ','DO ','DDS','DDM','DPM','CH ','OD ') THEN ECRED=CRED;

 IF ECRED IN ('MD ','DO ') THEN EPHYSTAT=1;
  ELSE IF ECRED IN ('DDS','DDM','DPM','CH ','OD ') THEN EPHYSTAT=2;
END;

IF EDOB=. THEN DO;
   SETDOBYR = SUBSTR(SETDOB,7,2);
   SDOBMMDD = SUBSTR(SETDOB,9,4);
```

```
   IF ' 5'<=SETDOBYR<='69' THEN DO;
      EDOBYR=SETDOBYR;
      IF ('0101'<=SDOBMMDD<='0131') ] ('0201'<=SDOBMMDD<='0228') ]
         (SDOBMMDD='0229' & (SETDOBYR/4=INT(SETDOBYR/4))) ]
         ('0301'<=SDOBMMDD<='0331') ] ('0401'<=SDOBMMDD<='0430') ]
         ('0501'<=SDOBMMDD<='0531') ] ('0601'<=SDOBMMDD<='0630') ]
         ('0701'<=SDOBMMDD<='0731') ] ('0801'<=SDOBMMDD<='0831') ]
         ('0901'<=SDOBMMDD<='0930') ] ('1001'<=SDOBMMDD<='1031') ]
         ('1101'<=SDOBMMDD<='1130') ] ('1201'<=SDOBMMDD<='1231')
         THEN EDOB=SETDOB;
   END;
  END;

  IF EGRADYR=. THEN DO;
   IF 30<=SETGRDYR<=93 THEN EGRADYR=SETGRDYR;
  END;
END;

RETAIN ESANC1 ESANCDAT ESANCYRS EMEDSCHL ECRED EPHYSTAT ELNAME
       EFNAME EMNAME ESUFFIX EDOB EDOBYR EDOD EDODYR EGRADYR;

DATA OUT.BASE(DROP=EDOBYR_N SETDATE);
       SET B;
       BY UPIN;

 /*****************************************************************/
 /* THE LAST RECORD PER UPIN TO PASS THROUGH THE FEASIBILITY TESTS */
 /* WILL HOLD THE COMPLETE "E" VARIABLE INFORMATION.               */
 /*****************************************************************/

IF LAST.UPIN;

 /*****************************************************************/
 /* WHEN YEAR OF BIRTH (EDOBYR) OR GRADUATION YEAR (EGRADYR) IS    */
 /* MISSING, THEY ARE IMPUTED BELOW. THE COUNT VARIABLES IDOBYR    */
 /* AND IGRADYR ARE CODED 0 WHEN EDOBYR & EGRADYR ARE BOTH MISSING, */
 /* 1 WHEN THEIR RESPECTIVE VARIABLE NEEDED IMPUTATION, OR MISSING */
 /* IF BOTH VARIABLES ARE INTACT.                                  */
 /*****************************************************************/

   IF EDOBYR='  ' & EGRADYR=. THEN DO;
    IGRADYR=0;
    IDOBYR=0;
   END;

   ELSE IF EDOBYR='  ' & EGRADYR^=. THEN DO;
    IF (30<=EGRADYR<=49) THEN GRADAGE=25;
    ELSE IF (50<=EGRADYR<=73) THEN GRADAGE=26;
    ELSE IF (74<=EGRADYR<=82) THEN GRADAGE=27;
```

```
    ELSE IF (83<=EGRADYR<=93) THEN GRADAGE=28;
    EDOBYR = EGRADYR - GRADAGE;
    IDOBYR = 1;
    IF EDOBYR<5 THEN DO;
       EDOBYR=.;
       IDOBYR=0;
    END;
   END;

   ELSE IF EDOBYR^='  ' & EGRADYR=. THEN DO;
    EDOBYR_N = EDOBYR*1;
    IF (' 5'<=EDOBYR<='59') THEN GRADAGE=27;
    ELSE IF ('60'<=EDOBYR<='65') THEN GRADAGE=26;
    ELSE IF ('66'<=EDOBYR<='69') THEN GRADAGE=25;
    EGRADYR = EDOBYR_N + GRADAGE;
    IGRADYR = 1;
    IF EGRADYR>93 THEN DO;
       EGRADYR=.;
       IGRADYR=0;
    END;
   END;

LABEL
      ELNAME    = 'Edited Physicians last name'
      EFNAME    = 'Edited Physicians first name'
      EMNAME    = 'Edited Physicians middle name'
      ESUFFIX   = 'Edited Physicians name suffix'
      EDOB      = 'Edited Date of birth (YYMMDD)'
      EDOBYR    = 'Edited Date of birth year(YY)'
      EDOD      = 'Edited Date of death (YYMMDD)'
      EDODYR    = 'Edited Date of death year(YY)'
      EMEDSCHL  = 'Edited medical school code'
      EGRADYR   = 'Edited Year phys. grad. from med. school'
      ESANC1    = 'Edited Sanction code physician is under'
      ESANCDAT  = 'Edited Sanction effective date (MMYY)'
      ESANCYRS  = 'Edited # years phys. has been sanctioned'
      EPHYSTAT  = 'Edited physician status'
      ECRED     = 'Edited physicians credentials'
      GRADAGE   = 'Age of physician at graduation'
      IDOBYR    = 'Date of birth year imputation indicator'
      IGRADYR   = 'Graduation year imputation indicator';
RUN;

PROC FREQ; TABLES ECRED EPHYSTAT EDODYR EDOBYR EGRADYR IGRADYR IDOBYR;
RUN;

PROC CONTENTS POSITION;
RUN;
```

APPENDIX B

SAS PROGRAM TO EDIT BUSINESS SETTING VARIABLES

```
 /*************************************************************/
 /*                                                           */
 /* -RESEARCHER EDITS: INDIVIDUAL VARIABLES                   */
 /*                    INDIVIDUAL BUSINESS SETTING RECORD     */
 /*                                                           */
 /*************************************************************/

DATA OUT.INDIVIDL
     (KEEP=UPIN ELICENSE ESTLICNS ESPEC1 ESPEC2 EBOARD1 EBOARD2 EPRCTIND
           EPRCTIND EPRT1IND DRIP CARRIER BILLZIP BIZZIP CPIN SETDATE);
    SET IN.JUL0293
     (KEEP=UPIN LICENSE STLICENS SPEC1 SPEC2 BOARD1 BOARD2 PRACTIND
           PRACTIND PART1IND INTERN SETDATE CARRIER MEDICARE BILLZIP
           BIZZIP SETDATE RENAME=(MEDICARE=CPIN));

/*****************************************/
/* BOGUS OBSERVATION REMOVED BY CODE BELOW */
/*****************************************/

IF SETDATE>0 THEN SETYR = SUBSTR(SETDATE,7,2);

IF SETYR='89' THEN SETYRADD=1;
ELSE IF SETYR='90' ] SETYR='91' THEN SETYRADD=2;
ELSE IF SETYR='92' ] SETYR='93' THEN SETYRADD=3;
ELSE DELETE;

/*************************************************************************/
/* FOLLOWING ARE EDITS ON THE INDIVIDUAL RECORD (PRACTICE SETTING).      */
/* IN THE CODE BELOW, "E" VARIABLES THAT NEED FEASIBILITY CHECKS         */
/* (LICENSE NUMBER, STATE LICENSED IN, PRIMARY & SECONDARY               */
/* SPECIALTY CODES, PRIMARY & SECONDARY BOARD CERTIFICATION VARIABLES,   */
/* GROUP PRACTICE INDICATOR, PHYSICIAN PARTICIPATION INDICATOR, DRIP     */
/* STATUS) ARE RUN THROUGH THEIR RESPECTIVE TESTS. THERE ARE 5 VARIABLES */
/* THAT ARE TAKEN AS IS (DATE SETTING ADDED, CARRIER NUMBER, THE MEDICARE */
/* BILLING NUMBER, THE BUSINESS & BILLING ZIPCODES).                     */
/*************************************************************************/

 IF LICENSE='000000000000' THEN ELICENSE='            ';
  ELSE ELICENSE=RIGHT(LICENSE);

 IF STLICENS IN ('AL','AK','AZ','AR','CA','CO','CT','DE','DC','FL',
                 'GA','HI','ID','IL','IN','IA','KS','KY','LA','ME',
                 'MD','MA','MI','MN','MS','MO','MT','NE','NV','NH',
                 'NJ','NM','NY','NC','ND','OH','OK','OR','PA','RI',
                 'SC','SD','TN','TX','UT','VT','VA','WA','WV','WI',
                 'WY','PR','GU','VI')
    THEN ESTLICNS=STLICENS;
```

```
 IF ('01'<=SPEC1<='41') OR (SPEC1 IN ('44','46','48','49','66','70','99'))
     OR ('76'<=SPEC1<='79') OR ('81'<=SPEC1<='86') OR ('90'<=SPEC1<='94')
     THEN ESPEC1=SPEC1;

 IF ('01'<=SPEC2<='41') OR (SPEC2 IN ('44','46','48','49','66','70','99'))
     OR ('76'<=SPEC2<='79') OR ('81'<=SPEC2<='86') OR ('90'<=SPEC2<='94')
     THEN ESPEC2=SPEC2;

 IF BOARD1 IN (' ','Y','N','U') THEN EBOARD1=BOARD1;

 IF BOARD2 IN (' ','Y','N','U') THEN EBOARD2=BOARD2;

 IF PRACTIND IN (1,4) THEN EPRCTIND=PRACTIND;

 IF PART1IND IN ('Y','N') THEN EPRT1IND=PART1IND;

 IF INTERN IN ('D','R','I','P') THEN DRIP=INTERN;

LABEL
      SETDATE   = 'Date the setting was added'
      ELICENSE  = 'Edited state license number'
      ESTLICNS  = 'Edited State physician is licensed'
      ESPEC1    = 'Edited Physicians primary specialty'
      EBOARD1   = 'Edited Primary spec code board cert ind'
      ESPEC2    = 'Edited Physicians secondary specialty'
      EBOARD2   = 'Edited Second spec code board cert ind'
      DRIP      = 'Edited The resident or interm indicator'
      EPRCTIND  = 'Edited The group pract or HMSA indicator'
      EPRT1IND  = 'Edited Phys participating indicator (1)';

RUN;

PROC CONTENTS POSITION;
```

APPENDIX C

SAS PROGRAM FOR LOGICAL CONSISTENCY EDITS

```
  /*************************************************************/
  /*                                                           */
  /* -RESEARCHER LOGICAL CONSISTENCY EDITS                     */
  /*                                                           */
  /*************************************************************/

  /*************************************************************/
  /* SORTING BASE RECORD BY MEDICAL SCHOOL IN ORDER TO MERGE */
  /* ON OPEN/CLOSE DATES                                       */
  /*************************************************************/

PROC SORT DATA=IN1.BASE OUT=M1; BY EMEDSCHL;

  /*********************************************************/
  /* MEDICAL SCHOOL OPEN & CLOSE DATE FILE                 */
  /*********************************************************/

DATA M2; INFILE IN3;
    INPUT EMEDSCHL $5. OPENYR CLOSEYR;
RUN;
PROC SORT; BY EMEDSCHL;
RUN;

  /*************************************************************/
  /* MERGING BASE & MEDICAL SHOOL OPEN/CLOSE FILE THEN       */
  /* SORTING BY UPIN                                           */
  /*************************************************************/

DATA M; MERGE M1(IN=INM1) M2;
        BY EMEDSCHL;
        IF INM1;
RUN;
PROC SORT; BY UPIN;
RUN;

  /************************************/
  /* MERGING BASE & INDIVIDUAL RECORD */
  /************************************/

DATA A;
     MERGE M
           IN2.INDIVIDL;
           BY UPIN;
```

```
/*************************************************************************/
/* EDITED PROVIDER CREDENTIALS (ECRED) & EDITED SCHOOL CODE (EMEDSCHL) */
/*************************************************************************/

IF ECRED IN ('MD ','DO ','DDM','DDS','DPM','CH ','OD ') THEN HCREDNTL='F';
ELSE HCREDNTL='M';

IF EMEDSCHL IN ('00102','00104','00106','00301','00401','00402',
               '00501','00502','00504','00505','00506','00507',
               '00508','00509','00511','00512','00513','00514',
               '00515','00516','00517','00518','00519','00576',
               '00702','00705','00801','00802','01001','01002',
               '01003','01102','01103','01104','01175','01201',
               '01205','01209','01211','01212','01218','01219',
               '01221','01222','01401','01601','01602','01604',
               '01605','01606','01608','01609','01610','01611',
               '01613','01614','01615','01616','01617','01618',
               '01619','01622','01623','01642','01643','01644',
               '01645','01675','01705','01708','01709','01717',
               '01718','01720','01801','01803','01804','01806',
               '01808','01810' '01875','01902','01903','01904',
               '01907','02001','02002','02004','02005','02007',
               '02008','02009','02011','02012','02101','02104',
               '02105','02106','02201','02275','02301','02303',
               '02304','02305','02306','02307','02308','02309',
               '02311','02312','02401','02405','02406','02407',
               '02415','02416','02501','02505','02507','02508',
               '02512','02576','02604','02605','02607','02608',
               '02701','02702','02801','02802','02803','02805',
               '02807','02808','02810','02820','02822','02826',
               '02828','02830','02833','02834','02835','02843',
               '02844','02845','02846','02878','02879','03004',
               '03005','03006','03007','03101','03201','03305',
               '03306','03375','03401','03501','03503','03506',
               '03508','03509','03510','03511','03513','03515',
               '03519','03520','03543','03545','03546','03547',
               '03548','03575','03601','03603','03604','03605',
               '03607','03608','03701','03801','03802','03803',
               '03806','03808','03809','03811','03819','03823',
               '03825','03826','03840','03841','03843','03844',
               '03845','03875','03901','03905','03979','04001',
               '04002','04101','04102','04107','04109','04111',
               '04112','04113','04114','04177','04201','04202',
               '04203','04204','04301','04501','04504','04601',
               '04701','04705','04706','04707','04708','04709',
               '04710','04711','04713','04714','04715','04720',
               '04802','04803','04804','04805','04806','04807',
               '04812','04813','04814','04815','04816','04878',
               '04901','05002','05101','05104','05106','05107',
```

```
              '05404','05415','05501','05502','05575','05602',
              '05603','05605','05606','000XF','000XG',
              '000XH','000XI','30000','30100','30200','30300',
              '30400','30500','30600','40000','40010','40020',
              '40030','40040','40050','40060','40070','40080',
              '40090','40100','40110','40120','40130','40140',
              '40150','04160','40170','50000','50010','50020',
              '50030','50040','50050','50060','50065','50070',
              '50080','50090','50100','50110','50120','50130',
              '50140','50150','50160','50170','50180','50190',
              '50200','50210','50220','50230','50240','50250',
              '50260','50270','50280','50290','50300','50310',
              '99999','10000','20000'
              '000XA','000XB','000XC','000XD','000XE','000XJ','000XK')
              THEN HMEDSCHL='F';
 ELSE HMEDSCHL='M';

IF HMEDSCHL='F' THEN DO;
  IF EMEDSCHL IN ('00576','01175','01675','01875','02275','02576',
                  '02878','02879','03375','03575','03875','03979'
                  '04177','04878','05575') THEN MSCHTYPE=7;
   ELSE IF ('30000'<=EMEDSCHL<='39999') ]
          (EMEDSCHL IN ('000XA','000XB','000XC','000XD','000XE',
                        '000XF','000XG','000XH','000XI','000XJ','000XK'))
                        THEN MSCHTYPE=3;
   ELSE IF EMEDSCHL<'10000' THEN MSCHTYPE=1;
   ELSE IF '10000'<=EMEDSCHL<='29999' THEN MSCHTYPE=2;
   ELSE IF '40000'<=EMEDSCHL<='49999' THEN MSCHTYPE=4;
   ELSE IF '50000'<=EMEDSCHL<='59999' THEN MSCHTYPE=5;
   ELSE IF EMEDSCHL='99999' THEN MSCHTYPE=6;
   ELSE MSCHTYPE=8;
END;

IF (MSCHTYPE=1 & ECRED='MD ') OR
   (MSCHTYPE=2 & (ECRED IN ('DDM','DDS'))) OR
   (MSCHTYPE=3 & ECRED='DPM') OR
   (MSCHTYPE=4 & ECRED='OD ') OR
   (MSCHTYPE=5 & ECRED='CH ') OR
   (MSCHTYPE=6 & (ECRED IN ('MD ','DDM','DDS','CH ','OD '))) OR
   (MSCHTYPE=7 & ECRED='DO ') THEN CSLCF='K';
 ELSE IF HMEDSCHL^='F' OR HCREDNTL^='F' THEN CSLCF='N';
 ELSE CSLCF='I';

/*****************************************/
/* ASSIGNMENT ALGORITHM (ECRED & EMEDSCHL) */
/*****************************************/

EHMDSCHL=HMEDSCHL;
EMSCHTYP=MSCHTYPE;
```

```
 IF CSLCF IN ('I','N') THEN DO;
    IF ECRED='MD ' THEN DO;
       EMEDSCHL='09098';
       EHMDSCHL='F';
       EMSCHTYP=1;
    END;
     ELSE IF ECRED='DO ' THEN DO;
       EMEDSCHL='09099';
       EHMDSCHL='F';
       EMSCHTYP=7;
     END;
     ELSE IF ECRED='DDM' THEN DO;
       EMEDSCHL='10000';
       EHMDSCHL='F';
       EMSCHTYP=2;
     END;
     ELSE IF ECRED='DDS' THEN DO;
       EMEDSCHL='20000';
       EHMDSCHL='F';
       EMSCHTYP=2;
     END;
     ELSE IF ECRED='DPM' THEN DO;
       EMEDSCHL='39999';
       EHMDSCHL='F';
       EMSCHTYP=3;
     END;
     ELSE IF ECRED='OD ' THEN DO;
       EMEDSCHL='49999';
       EHMDSCHL='F';
       EMSCHTYP=4;
     END;
     ELSE IF ECRED='CH ' THEN DO;
       EMEDSCHL='50000';
       EHMDSCHL='F';
       EMSCHTYP=5;
     END;
 END;

 /*************************/
 /* BASE RECORD PROCESSING */
 /*************************/

IF FIRST.UPIN THEN DO;

 BASE=1;

 IF (EMSCHTYP=1 & ECRED='MD ') OR
    (EMSCHTYP=2 & (ECRED IN ('DDM','DDS'))) OR
    (EMSCHTYP=3 & ECRED='DPM') OR
    (EMSCHTYP=4 & ECRED='OD ') OR
```

```
   (EMSCHTYP=5 & ECRED='CH ') OR
   (EMSCHTYP=6 & (ECRED IN ('MD ','DDM','DDS','CH ','OD '))) OR
   (EMSCHTYP=7 & ECRED='DO ') THEN ECSLCF='K';
 ELSE IF EHMDSCHL^='F' OR HCREDNTL^='F' THEN ECSLCF='N';
 ELSE ECSLCF='I';

/*****************************************************/
/* BIRTH YEAR (EDOBYR) AND GRADUATION YEAR (EGRADYR) */
/*****************************************************/

IF ' 5'<=EDOBYR<='69' THEN BYF='F';
IF 30<=EGRADYR<=93 THEN GYF='F';

/*****************************************************/
/* GRADUATION AGE (GRADAGE) CALCULATION & IMPUTATION */
/*****************************************************/

IF BYF='F' & GYF='F' THEN DO;
   BGLCF='K';
   GRADAGE = EGRADYR - EDOBYR;
   IF GRADAGE<18 OR GRADAGE>40 THEN DO;
      IF ' 5'<=EDOBYR<='59' THEN GRADAGE=27;
      ELSE IF '60'<=EDOBYR<='65' THEN GRADAGE=26;
      ELSE IF '66'<=EDOBYR<='69' THEN GRADAGE=25;
      EGRADYR=EDOBYR + GRADAGE;
      IF EGRADYR>93 THEN DO;
         EGRADYR=.;
         IGRADYR=0;
         BGLCF='N';
      END;
   END;
END;
ELSE BGLCF='N';

/*************************************************************************/
/* GRADUATION YEAR (EGRADYR) & MED SCHOOL OPEN/CLOSE DATES (OPEN,CLOSE)  */
/* & MED SCHOOL TYPE (EMSCHTYP)                                          */
/*************************************************************************/

IF EMSCHTYP^=1 THEN GMLCF=' ';
 ELSE IF EMEDSCHL='09098' OR BGLCF^='K' THEN GMLCF='N';
   ELSE IF EMEDSCHL='05415' THEN GMLCF='I';
    ELSE DO;
      FULLGRAD=1900+EGRADYR;
      IF OPENYR<=FULLGRAD<=CLOSEYR THEN GMLCF='K';
       ELSE IF FULLGRAD<OPENYR THEN DO;
        IF OPENYR - FULLGRAD <= 12 THEN DO;
         EGRADYR=(OPENYR-1900) + 2;
         GRADAGE=EGRADYR - EDOBYR;
         GMLCF='K';
```

```
        END;
        ELSE DO;
         EGRADYR=.;
         GRADAGE=.;
         GMLCF='N';
        END;
       END;
       ELSE DO;
        GYROVR = FULLGRAD - CLOSEYR;
        IF GRADAGE-GYROVR >= 18 THEN DO;
         EGRADYR = CLOSEYR - 1900;
         GRADAGE = EGRADYR - EDOBYR;
         GMLCF='K';
        END;
        ELSE DO;
         EGRADYR=.;
         GRADAGE=.;
         GMLCF='N';
        END;
       END;
   END;
END;

/*********************************/
/* INDIVIDUAL RECORD PROCESSING */
/*********************************/
/*************************************************************/
/* PROVIDER CREDENTIALS (ECRED) & SPECIALTY CODE (ESPEC1)    */
/*************************************************************/

IF ('01'<=ESPEC1<='41') OR (ESPEC1 IN ('44','46','48','66','70')) OR
   ('76'<=ESPEC1<='79') OR ('81'<=ESPEC1<='86') OR ('90'<=ESPEC1<='94')
   THEN HSPEC1='F';
ELSE IF ESPEC1 IN ('  ',' 0','00') THEN HSPEC1='M';
ELSE IF ESPEC1='99' THEN HSPEC1='U';
ELSE IF ESPEC1='49' THEN HSPEC1='S';
ELSE HSPEC1='E';

IF ESPEC1 IN
           ('01','02','03','04','05','06','07','08','10','11','13','14',
            '16','18','20','22','24','25','26','28','29','30','33',
            '34','36','37','38','39','40','44','46','66',
            '70','76','77','78','79','81','82','83','84','86','90',
            '91','92','93','94')
            THEN SPECS1=1;
 ELSE IF ESPEC1 IN
           ('09','12','15','17','21','23','27','31','32') THEN SPECS1=2;
 ELSE IF ESPEC1='19' THEN SPECS1=3;
 ELSE IF ESPEC1='85' THEN SPECS1=4;
 ELSE IF ESPEC1='35' THEN SPECS1=5;
```

```
 ELSE IF ESPEC1='48' THEN SPECS1=6;
 ELSE IF ESPEC1='41' THEN SPECS1=7;
 ELSE IF ESPEC1='49' THEN SPECS1=8;
 ELSE IF ESPEC1='99' THEN SPECS1=9;
 ELSE SPECS1=10;

IF (ECRED='MD ' & (SPECS1 IN (1,8,9))) OR
   (ECRED='DO ' & (SPECS1 IN (1,2,8,9))) OR
   ((ECRED IN ('DDM','DDS')) & (SPECS1 IN (3,4))) OR
   (ECRED='CH ' & SPECS1=5) OR
   (ECRED='DPM' & SPECS1=6) OR
   (ECRED='OD ' & SPECS1=7) THEN CPLCF='K';
 ELSE IF HCREDNTL^='F' OR (HSPEC1 NOT IN ('F','S','U')) THEN CPLCF='N';
 ELSE CPLCF='I';

/*****************************************/
/* ASSIGNMENT ALGORITHM (ECRED & EESPEC1) */
/*****************************************/

ESPECS1=SPECS1;
EHSPEC1=HSPEC1;

IF CPLCF IN ('I','N') THEN DO;
   IF ECRED IN ('MD ','DO ') THEN DO;
      ESPEC1='99';
      ESPECS1=9;
      EHSPEC1='U';
   END;
    ELSE IF ECRED IN ('DDM','DDS') THEN DO;
      ESPEC1='19';
      ESPECS1=3;
      EHSPEC1='F';
    END;
    ELSE IF ECRED='CH ' THEN DO;
      ESPEC1='35';
      ESPECS1=5;
      EHSPEC1='F';
    END;
    ELSE IF ECRED='DPM' THEN DO;
      ESPEC1='48';
      ESPECS1=6;
      EHSPEC1='F';
    END;
    ELSE IF ECRED='OD ' THEN DO;
      ESPEC1='41';
      ESPECS1=7;
      EHSPEC1='F';
    END;
END;
```

```
IF (ECRED='MD ' & (ESPECS1 IN (1,8,9))) OR
   (ECRED='DO ' & (ESPECS1 IN (1,2,8,9))) OR
   ((ECRED IN ('DDM','DDS')) & (ESPECS1 IN (3,4))) OR
   (ECRED='CH ' & ESPECS1=5) OR
   (ECRED='DPM' & ESPECS1=6) OR
   (ECRED='OD ' & ESPECS1=7) THEN ECPLCF='K';
 ELSE IF HCREDNTL^='F' OR (EHSPEC1 NOT IN ('F','S','U')) THEN ECPLCF='N';
 ELSE ECPLCF='I';

/*****************************************/
/* ASSIGNMENT ALGORITHM (ECRED & ESPEC1B)  */
/*****************************************/

ESPEC1B=ESPEC1;
ESPECS1B=ESPECS1;
EHSPEC1B=EHSPEC1;

IF ECRED='DO ' THEN DO;
   ESPECS1B=1;
   IF ESPEC1 IN ('09','15') THEN ESPEC1B=16;
    ELSE IF ESPEC1='17' THEN ESPEC1B='04';
    ELSE IF ESPEC1='21' THEN ESPEC1B='22';
    ELSE IF ESPEC1='23' THEN ESPEC1B='76';
    ELSE IF ESPEC1='27' THEN ESPEC1B='86';
    ELSE IF ESPEC1='31' THEN ESPEC1B='30';
    ELSE IF ESPEC1='32' THEN ESPEC1B='92';
END;

IF (ECRED='MD ' & (ESPECS1B IN (1,8,9))) OR
   (ECRED='DO ' & (ESPECS1B IN (1,2,8,9))) OR
   ((ECRED IN ('DDM','DDS')) & (ESPECS1B IN (3,4))) OR
   (ECRED='CH ' & ESPECS1B=5) OR
   (ECRED='DPM' & ESPECS1B=6) OR
   (ECRED='OD ' & ESPECS1B=7) THEN ECPLCFB='K';
 ELSE IF HCREDNTL^='F' OR (EHSPEC1B NOT IN ('F','S','U')) THEN ECPLCFB='N';
 ELSE ECPLCFB='I';
```

```
/*************************************************************/
/* SCHOOL TYPE (EMSCHTYP) AND PRIMARY SPECIALTY TYPE (ESPECS1) */
/*************************************************************/

  IF (EMSCHTYP=1 & (ESPECS1 IN (1,8,9))) OR
     (EMSCHTYP=2 & (ESPECS1 IN (3,4))) OR
     (EMSCHTYP=3 & ESPECS1=6) OR
     (EMSCHTYP=4 & ESPECS1=7) OR
     (EMSCHTYP=5 & ESPECS1=5) OR
     (EMSCHTYP=6 & (ESPECS1 IN (1,3,4,5,7,8,9))) OR
     (EMSCHTYP=7 & (ESPECS1 IN (1,2,8,9))) THEN SPLCF='K';
   ELSE IF EHMDSCHL^='F' OR (EHSPEC1 NOT IN ('F','S','U'))
    THEN SPLCF='N';
   ELSE SPLCF='I';

RUN;

PROC SORT DATA=A
 OUT=OUT.EDIT01
     (DROP=BASE IGRADYR IDOBYR OPENYR CLOSEYR HCREDNTL HMEDSCHL MSCHTYPE
           CSLCF EHMDSCHL EMSCHTYP ECSLCF BYF GYF BGLCF GMLCF FULLGRAD
           GYROVR HSPEC1 SPECS1 CPLCF ESPECS1 EHSPEC1 ECPLCF ESPECS1B
           EHSPEC1B ECPLCFB SPLCF);
 BY UPIN;
RUN;
```

APPENDIX D

SAS PROGRAMS FOR DETECTING MULTIPLE-UPIN PROVIDERS

APPENDIX D-1

ID_KEY_1 = DOB,L1_3,F1,LNAMESUM

```
  /*********************************************************/
  /*                                                       */
  /* -CHECK FOR MULTIPLE UPINS PER PHYSICIAN FOR ACTIVE    */
  /*  PRACTICE SETTINGS.                                   */
  /*                                                       */
  /*********************************************************/

DATA A;
     SET IN.JUL0293
     (WHERE=(DOB>0)
      KEEP=UPIN INTERN SETDATE DOB LNAME FNAME);

  LNAME=LEFT(LNAME);
  FNAME=LEFT(FNAME);

 /*****************************************/
 /* BOGUS OBSERVATION REMOVED BY CODE BELOW */
 /*****************************************/

 IF SETDATE>0 THEN SETYR = SUBSTR(SETDATE,7,2);

 IF SETYR='89' THEN SETYRADD=1;
 ELSE IF SETYR='90' ] SETYR='91' THEN SETYRADD=2;
 ELSE IF SETYR='92' ] SETYR='93' THEN SETYRADD=3;
 ELSE DELETE;

 /*****************************************************************/
 /* ACTIVE (OR MISSING) PRACTICE SETTINGS SORTED BY UPIN & SETDATE */
 /*****************************************************************/

IF (INTERN='R' ] INTERN='I' ] INTERN='P') THEN DRIP='A';
 ELSE IF INTERN='D' THEN DRIP='D';
 ELSE DRIP='M';

IF DRIP IN ('A','M');

RUN;

PROC SORT DATA=A; BY UPIN SETDATE;
RUN;

DATA B(KEEP=UPIN ID_KEY_1 MU_FL_1)
     C(KEEP=UPIN MU_FL_1);
       SET A(KEEP=UPIN DOB LNAME FNAME);
        BY UPIN;
        IF FIRST.UPIN;

INFORMAT MU_FL_1 1.;
```

```
 /*************************/
 /* CODE FOR FEASIBLE DOB */
 /*************************/

  DOBYR = SUBSTR(DOB,7,2);
  DOBMO = SUBSTR(DOB,9,2);
  MODAY = SUBSTR(DOB,9,4);
  IF DOBYR='  ' THEN DO;
   IF (101<=DOB<=131) ] (201<=DOB<=229) ]
      (301<=DOB<=331) ] (401<=DOB<=430) ]
      (501<=DOB<=531) ] (601<=DOB<=630) ]
      (701<=DOB<=731) ] (801<=DOB<=831) ]
      (901<=DOB<=930) ] (1001<=DOB<=1031) ]
      (1101<=DOB<=1130) ] (1201<=DOB<=1231)
      THEN DOBYR='00';
   END;

IF DOBYR='00' THEN BYRH=0;
ELSE IF ' 1'<=DOBYR<'20' THEN BYRH=1;
ELSE IF '20'<=DOBYR<='69' THEN BYRH=2;
ELSE IF '70'<=DOBYR<='98' THEN BYRH=3;
ELSE IF DOBYR='99' & ('  '<DOBMO<'13') THEN BYRH=4;
ELSE IF DOBYR='99' THEN BYRH=5;
ELSE IF DOB=0 THEN BYRH=6;

IF BYRH=. ] BYRH=5 ] BYRH=6 THEN BYRHX=1;
ELSE IF BYRH=3 THEN BYRHX=2;
ELSE IF BYRH=4 THEN BYRHX=3;
ELSE BYRHX=4;

IF BYRHX=4 & MODAY^='0000' & DOB^=10101 &
    (('  101'<=MODAY<='0131') ] (' 201'<=MODAY<='0229') ]
     (' 301'<=MODAY<='0331') ] (' 401'<=MODAY<='0430') ]
     (' 501'<=MODAY<='0531') ] (' 601'<=MODAY<='0630') ]
     (' 701'<=MODAY<='0731') ] (' 801'<=MODAY<='0831') ]
     (' 901'<=MODAY<='0930') ] ('1001'<=MODAY<='1031') ]
     ('1101'<=MODAY<='1130') ] ('1201'<=MODAY<='1231')) THEN DO;

 /*************************************************************************/
 /* IF THERE IS A FEASIBLE DATE OF BIRTH, PASS THROUGH ID_KEY ALGORITHM */
 /*************************************************************************/

 INFORMAT F1 $1. L1-L20 $1. L1_3 $3. LN1-LN20 2.;

 L1 = SUBSTR(LNAME,1,1);
 L2 = SUBSTR(LNAME,2,1);
 L3 = SUBSTR(LNAME,3,1);
 L4 = SUBSTR(LNAME,4,1);
 L5 = SUBSTR(LNAME,5,1);
 L6 = SUBSTR(LNAME,6,1);
```

```
L7 = SUBSTR(LNAME,7,1);
L8 = SUBSTR(LNAME,8,1);
L9 = SUBSTR(LNAME,9,1);
L10 = SUBSTR(LNAME,10,1);
L11 = SUBSTR(LNAME,11,1);
L12 = SUBSTR(LNAME,12,1);
L13 = SUBSTR(LNAME,13,1);
L14 = SUBSTR(LNAME,14,1);
L15 = SUBSTR(LNAME,15,1);
L16 = SUBSTR(LNAME,16,1);
L17 = SUBSTR(LNAME,17,1);
L18 = SUBSTR(LNAME,18,1);
L19 = SUBSTR(LNAME,19,1);
L20 = SUBSTR(LNAME,20,1);

IF ('A'<=L1<='Z') & ('A'<=L2<='Z') & ('A'<=L3<='Z') THEN
    L1_3 = L1||L2||L3;
ELSE IF ('A'<=L1<='Z') & ('A'<=L2<='Z') & ('A'<=L4<='Z') THEN
    L1_3 = L1||L2||L4;
ELSE IF ('A'<=L1<='Z') & ('A'<=L2<='Z') & ('A'<=L5<='Z') THEN
    L1_3 = L1||L2||L5;
ELSE IF ('A'<=L1<='Z') & ('A'<=L3<='Z') & ('A'<=L4<='Z') THEN
    L1_3 = L1||L3||L4;
ELSE IF ('A'<=L1<='Z') & ('A'<=L4<='Z') & ('A'<=L5<='Z') THEN
    L1_3 = L1||L4||L5;
ELSE IF ('A'<=L2<='Z') & ('A'<=L3<='Z') & ('A'<=L4<='Z') THEN
    L1_3 = L2||L3||L4;
ELSE IF ('A'<=L2<='Z') & ('A'<=L3<='Z') & ('A'<=L5<='Z') THEN
    L1_3 = L2||L3||L5;
ELSE IF ('A'<=L3<='Z') & ('A'<=L4<='Z') & ('A'<=L5<='Z') THEN
    L1_3 = L3||L4||L5;
ELSE L1_3 = L1||L2||L3;

F1 = SUBSTR(FNAME,1,1);

ARRAY L(20) L1-L20;
ARRAY LN(20) LN1-LN20;

  DO I = 1 TO 20;
     IF L(I) IN ('A','E','I','O','U') THEN LN(I)=0;
     ELSE IF L(I)='B' THEN LN(I)=1;
     ELSE IF L(I)='C' THEN LN(I)=2;
     ELSE IF L(I)='D' THEN LN(I)=3;
     ELSE IF L(I)='F' THEN LN(I)=4;
     ELSE IF L(I)='G' THEN LN(I)=5;
     ELSE IF L(I)='H' THEN LN(I)=6;
     ELSE IF L(I)='J' THEN LN(I)=7;
     ELSE IF L(I)='K' THEN LN(I)=8;
     ELSE IF L(I)='L' THEN LN(I)=9;
     ELSE IF L(I)='M' THEN LN(I)=10;
```

```
      ELSE IF L(I)='N' THEN LN(I)=11;
      ELSE IF L(I)='P' THEN LN(I)=12;
      ELSE IF L(I)='Q' THEN LN(I)=13;
      ELSE IF L(I)='R' THEN LN(I)=14;
      ELSE IF L(I)='S' THEN LN(I)=15;
      ELSE IF L(I)='T' THEN LN(I)=16;
      ELSE IF L(I)='V' THEN LN(I)=17;
      ELSE IF L(I)='W' THEN LN(I)=18;
      ELSE IF L(I)='X' THEN LN(I)=19;
      ELSE IF L(I)='Y' THEN LN(I)=20;
      ELSE IF L(I)='Z' THEN LN(I)=21;
     END;

 LNAMESUM = SUM(OF LN1-LN20);

 ID_KEY_1 = DOB]]L1_3]]F1]]LNAMESUM;

 OUTPUT B;

END;

 /*****************************************************************/
 /* IF THERE IS NOT A FEASIBLE DATE OF BIRTH, ASSIGN FLAG VALUE */
 /*****************************************************************/

ELSE DO;
 MU_FL_1=2;
 OUTPUT C;
END;

RUN;

PROC SORT DATA=B; BY ID_KEY_1;
RUN;

DATA B1; SET B;
        BY ID_KEY_1;

 /*********************************/
 /* UNIQUE ID_KEY FLAG ASSIGNMENT */
 /*********************************/

IF FIRST.ID_KEY_1 & LAST.ID_KEY_1 THEN MU_FL_1=3;

 /*************************************/
 /* MULTIPLE ID_KEY FLAG ASSIGNMENT */
 /*************************************/

 ELSE MU_FL_1=4;
RUN;
```

```
PROC SORT DATA=B1; BY UPIN;
RUN;

 /**************************/
 /* PUT DATA BACK TOGETHER */
 /**************************/

DATA OUT.KEY1; SET B1 C;
          BY UPIN;
RUN;

PROC FREQ DATA=OUT.KEY1; TABLES MU_FL_1;
RUN;

PROC CONTENTS POSITION DATA=OUT.KEY1;
RUN;
```

APPENDIX D-2

ID_KEY_2 = DOB,L1_3,LNAMESUM

```
  /*******************************************************/
  /*                                                     */
  /* -CHECK FOR MULTIPLE UPINS PER PHYSICIAN FOR ACTIVE  */
  /*  PRACTICE SETTINGS.                                 */
  /*                                                     */
  /*******************************************************/

DATA A;
     SET IN.JUL0293
     (WHERE=(DOB>0)
      KEEP=UPIN INTERN SETDATE DOB LNAME);

  LNAME=LEFT(LNAME);

 /*****************************************/
 /* BOGUS OBSERVATION REMOVED BY CODE BELOW */
 /*****************************************/

 IF SETDATE>0 THEN SETYR = SUBSTR(SETDATE,7,2);

 IF SETYR='89' THEN SETYRADD=1;
 ELSE IF SETYR='90' ] SETYR='91' THEN SETYRADD=2;
 ELSE IF SETYR='92' ] SETYR='93' THEN SETYRADD=3;
 ELSE DELETE;

 /*****************************************************************/
 /* ACTIVE (OR MISSING) PRACTICE SETTINGS SORTED BY UPIN & SETDATE */
 /*****************************************************************/

IF (INTERN='R' ] INTERN='I' ] INTERN='P') THEN DRIP='A';
 ELSE IF INTERN='D' THEN DRIP='D';
 ELSE DRIP='M';

IF DRIP IN ('A','M');

RUN;

PROC SORT DATA=A; BY UPIN SETDATE;
RUN;

DATA B(KEEP=UPIN ID_KEY_2 MU_FL_2)
     C(KEEP=UPIN MU_FL_2);
       SET A(KEEP=UPIN DOB LNAME);
        BY UPIN;
        IF FIRST.UPIN;

INFORMAT MU_FL_2 1.;
```

```
 /*************************/
 /* CODE FOR FEASIBLE DOB */
 /*************************/

  DOBYR = SUBSTR(DOB,7,2);
  DOBMO = SUBSTR(DOB,9,2);
  MODAY = SUBSTR(DOB,9,4);
  IF DOBYR='  ' THEN DO;
   IF (101<=DOB<=131) ] (201<=DOB<=229) ]
      (301<=DOB<=331) ] (401<=DOB<=430) ]
      (501<=DOB<=531) ] (601<=DOB<=630) ]
      (701<=DOB<=731) ] (801<=DOB<=831) ]
      (901<=DOB<=930) ] (1001<=DOB<=1031) ]
      (1101<=DOB<=1130) ] (1201<=DOB<=1231)
      THEN DOBYR='00';
   END;

IF DOBYR='00' THEN BYRH=0;
ELSE IF ' 1'<=DOBYR<'20' THEN BYRH=1;
ELSE IF '20'<=DOBYR<='69' THEN BYRH=2;
ELSE IF '70'<=DOBYR<='98' THEN BYRH=3;
ELSE IF DOBYR='99' & ('  '<DOBMO<'13') THEN BYRH=4;
ELSE IF DOBYR='99' THEN BYRH=5;
ELSE IF DOB=0 THEN BYRH=6;

IF BYRH=. ] BYRH=5 ] BYRH=6 THEN BYRHX=1;
ELSE IF BYRH=3 THEN BYRHX=2;
ELSE IF BYRH=4 THEN BYRHX=3;
ELSE BYRHX=4;

IF BYRHX=4 & MODAY^='0000' & DOB^=10101 &
    (('  101'<=MODAY<='0131') ] (' 201'<=MODAY<='0229') ]
     (' 301'<=MODAY<='0331') ] (' 401'<=MODAY<='0430') ]
     (' 501'<=MODAY<='0531') ] (' 601'<=MODAY<='0630') ]
     (' 701'<=MODAY<='0731') ] (' 801'<=MODAY<='0831') ]
     (' 901'<=MODAY<='0930') ] ('1001'<=MODAY<='1031') ]
     ('1101'<=MODAY<='1130') ] ('1201'<=MODAY<='1231')) THEN DO;

 /*************************************************************************/
 /* IF THERE IS A FEASIBLE DATE OF BIRTH, PASS THROUGH ID_KEY ALGORITHM */
 /*************************************************************************/

 INFORMAT L1-L20 $1. L1_3 $3. LN1-LN20 2.;

 L1 = SUBSTR(LNAME,1,1);
 L2 = SUBSTR(LNAME,2,1);
 L3 = SUBSTR(LNAME,3,1);
 L4 = SUBSTR(LNAME,4,1);
 L5 = SUBSTR(LNAME,5,1);
 L6 = SUBSTR(LNAME,6,1);
```

```
L7 = SUBSTR(LNAME,7,1);
L8 = SUBSTR(LNAME,8,1);
L9 = SUBSTR(LNAME,9,1);
L10 = SUBSTR(LNAME,10,1);
L11 = SUBSTR(LNAME,11,1);
L12 = SUBSTR(LNAME,12,1);
L13 = SUBSTR(LNAME,13,1);
L14 = SUBSTR(LNAME,14,1);
L15 = SUBSTR(LNAME,15,1);
L16 = SUBSTR(LNAME,16,1);
L17 = SUBSTR(LNAME,17,1);
L18 = SUBSTR(LNAME,18,1);
L19 = SUBSTR(LNAME,19,1);
L20 = SUBSTR(LNAME,20,1);

IF ('A'<=L1<='Z') & ('A'<=L2<='Z') & ('A'<=L3<='Z') THEN
    L1_3 = L1]]L2]]L3;
ELSE IF ('A'<=L1<='Z') & ('A'<=L2<='Z') & ('A'<=L4<='Z') THEN
    L1_3 = L1]]L2]]L4;
ELSE IF ('A'<=L1<='Z') & ('A'<=L2<='Z') & ('A'<=L5<='Z') THEN
    L1_3 = L1]]L2]]L5;
ELSE IF ('A'<=L1<='Z') & ('A'<=L3<='Z') & ('A'<=L4<='Z') THEN
    L1_3 = L1]]L3]]L4;
ELSE IF ('A'<=L1<='Z') & ('A'<=L4<='Z') & ('A'<=L5<='Z') THEN
    L1_3 = L1]]L4]]L5;
ELSE IF ('A'<=L2<='Z') & ('A'<=L3<='Z') & ('A'<=L4<='Z') THEN
    L1_3 = L2]]L3]]L4;
ELSE IF ('A'<=L2<='Z') & ('A'<=L3<='Z') & ('A'<=L5<='Z') THEN
    L1_3 = L2]]L3]]L5;
ELSE IF ('A'<=L3<='Z') & ('A'<=L4<='Z') & ('A'<=L5<='Z') THEN
    L1_3 = L3]]L4]]L5;
ELSE L1_3 = L1]]L2]]L3;

ARRAY L(20) L1-L20;
ARRAY LN(20) LN1-LN20;

  DO I = 1 TO 20;
     IF L(I) IN ('A','E','I','O','U') THEN LN(I)=0;
     ELSE IF L(I)='B' THEN LN(I)=1;
     ELSE IF L(I)='C' THEN LN(I)=2;
     ELSE IF L(I)='D' THEN LN(I)=3;
     ELSE IF L(I)='F' THEN LN(I)=4;
     ELSE IF L(I)='G' THEN LN(I)=5;
     ELSE IF L(I)='H' THEN LN(I)=6;
     ELSE IF L(I)='J' THEN LN(I)=7;
     ELSE IF L(I)='K' THEN LN(I)=8;
     ELSE IF L(I)='L' THEN LN(I)=9;
     ELSE IF L(I)='M' THEN LN(I)=10;
     ELSE IF L(I)='N' THEN LN(I)=11;
     ELSE IF L(I)='P' THEN LN(I)=12;
```

```
      ELSE IF L(I)='Q' THEN LN(I)=13;
      ELSE IF L(I)='R' THEN LN(I)=14;
      ELSE IF L(I)='S' THEN LN(I)=15;
      ELSE IF L(I)='T' THEN LN(I)=16;
      ELSE IF L(I)='V' THEN LN(I)=17;
      ELSE IF L(I)='W' THEN LN(I)=18;
      ELSE IF L(I)='X' THEN LN(I)=19;
      ELSE IF L(I)='Y' THEN LN(I)=20;
      ELSE IF L(I)='Z' THEN LN(I)=21;
     END;

 LNAMESUM = SUM(OF LN1-LN20);

 ID_KEY_2 = DOB]]L1_3]]LNAMESUM;

 OUTPUT B;

END;

 /*****************************************************************/
 /* IF THERE IS NOT A FEASIBLE DATE OF BIRTH, ASSIGN FLAG VALUE */
 /*****************************************************************/

ELSE DO;
 MU_FL_2=2;
 OUTPUT C;
END;

RUN;

PROC SORT DATA=B; BY ID_KEY_2;
RUN;

DATA B1; SET B;
        BY ID_KEY_2;

 /*********************************/
 /* UNIQUE ID_KEY FLAG ASSIGNMENT */
 /*********************************/

IF FIRST.ID_KEY_2 & LAST.ID_KEY_2 THEN MU_FL_2=3;

 /************************************/
 /* MULTIPLE ID_KEY FLAG ASSIGNMENT */
 /************************************/

 ELSE MU_FL_2=4;
RUN;

PROC SORT DATA=B1; BY UPIN;
```

```
RUN;

 /*************************/
 /* PUT DATA BACK TOGETHER */
 /*************************/

DATA OUT.KEY2; SET B1 C;
          BY UPIN;
RUN;

PROC FREQ DATA=OUT.KEY2; TABLES MU_FL_2;
RUN;

PROC CONTENTS POSITION DATA=OUT.KEY2;
RUN;
```

APPENDIX D-3

ID_KEY_3 = DOB,L1_3,F1

```
  /*********************************************************/
  /*                                                       */
  /* -CHECK FOR MULTIPLE UPINS PER PHYSICIAN FOR ACTIVE    */
  /*  PRACTICE SETTINGS.                                   */
  /*                                                       */
  /*********************************************************/

DATA A;
     SET IN.JUL0293
     (WHERE=(DOB>0)
      KEEP=UPIN INTERN SETDATE DOB LNAME FNAME);

  LNAME=LEFT(LNAME);
  FNAME=LEFT(FNAME);

 /*****************************************/
 /* BOGUS OBSERVATION REMOVED BY CODE BELOW */
 /*****************************************/

 IF SETDATE>0 THEN SETYR = SUBSTR(SETDATE,7,2);

 IF SETYR='89' THEN SETYRADD=1;
 ELSE IF SETYR='90' ] SETYR='91' THEN SETYRADD=2;
 ELSE IF SETYR='92' ] SETYR='93' THEN SETYRADD=3;
 ELSE DELETE;

 /*****************************************************************/
 /* ACTIVE (OR MISSING) PRACTICE SETTINGS SORTED BY UPIN & SETDATE */
 /*****************************************************************/

IF (INTERN='R' ] INTERN='I' ] INTERN='P') THEN DRIP='A';
 ELSE IF INTERN='D' THEN DRIP='D';
 ELSE DRIP='M';

IF DRIP IN ('A','M');

RUN;

PROC SORT DATA=A; BY UPIN SETDATE;
RUN;

DATA B(KEEP=UPIN ID_KEY_3 MU_FL_3)
     C(KEEP=UPIN MU_FL_3);
       SET A(KEEP=UPIN DOB LNAME FNAME);
        BY UPIN;
        IF FIRST.UPIN;

INFORMAT MU_FL_3 1.;
```

```
/*************************/
/* CODE FOR FEASIBLE DOB */
/*************************/

 DOBYR = SUBSTR(DOB,7,2);
 DOBMO = SUBSTR(DOB,9,2);
 MODAY = SUBSTR(DOB,9,4);
 IF DOBYR='  ' THEN DO;
  IF (101<=DOB<=131) ] (201<=DOB<=229) ]
     (301<=DOB<=331) ] (401<=DOB<=430) ]
     (501<=DOB<=531) ] (601<=DOB<=630) ]
     (701<=DOB<=731) ] (801<=DOB<=831) ]
     (901<=DOB<=930) ] (1001<=DOB<=1031) ]
     (1101<=DOB<=1130) ] (1201<=DOB<=1231)
     THEN DOBYR='00';
  END;

IF DOBYR='00' THEN BYRH=0;
ELSE IF ' 1'<=DOBYR<'20' THEN BYRH=1;
ELSE IF '20'<=DOBYR<='69' THEN BYRH=2;
ELSE IF '70'<=DOBYR<='98' THEN BYRH=3;
ELSE IF DOBYR='99' & ('  '<DOBMO<'13') THEN BYRH=4;
ELSE IF DOBYR='99' THEN BYRH=5;
ELSE IF DOB=0 THEN BYRH=6;

IF BYRH=. ] BYRH=5 ] BYRH=6 THEN BYRHX=1;
ELSE IF BYRH=3 THEN BYRHX=2;
ELSE IF BYRH=4 THEN BYRHX=3;
ELSE BYRHX=4;

IF BYRHX=4 & MODAY^='0000' & DOB^=10101 &
    (('  101'<=MODAY<='0131') ] (' 201'<=MODAY<='0229') ]
     (' 301'<=MODAY<='0331') ] (' 401'<=MODAY<='0430') ]
     (' 501'<=MODAY<='0531') ] (' 601'<=MODAY<='0630') ]
     (' 701'<=MODAY<='0731') ] (' 801'<=MODAY<='0831') ]
     (' 901'<=MODAY<='0930') ] ('1001'<=MODAY<='1031') ]
     ('1101'<=MODAY<='1130') ] ('1201'<=MODAY<='1231')) THEN DO;

 /*************************************************************************/
 /* IF THERE IS A FEASIBLE DATE OF BIRTH, PASS THROUGH ID_KEY ALGORITHM */
 /*************************************************************************/

 INFORMAT F1 $1. L1-L5 $1. L1_3 $3.;

 L1 = SUBSTR(LNAME,1,1);
 L2 = SUBSTR(LNAME,2,1);
 L3 = SUBSTR(LNAME,3,1);
 L4 = SUBSTR(LNAME,4,1);
 L5 = SUBSTR(LNAME,5,1);
```

```
 IF ('A'<=L1<='Z') & ('A'<=L2<='Z') & ('A'<=L3<='Z') THEN
     L1_3 = L1]]L2]]L3;
 ELSE IF ('A'<=L1<='Z') & ('A'<=L2<='Z') & ('A'<=L4<='Z') THEN
     L1_3 = L1]]L2]]L4;
 ELSE IF ('A'<=L1<='Z') & ('A'<=L2<='Z') & ('A'<=L5<='Z') THEN
     L1_3 = L1]]L2]]L5;
 ELSE IF ('A'<=L1<='Z') & ('A'<=L3<='Z') & ('A'<=L4<='Z') THEN
     L1_3 = L1]]L3]]L4;
 ELSE IF ('A'<=L1<='Z') & ('A'<=L4<='Z') & ('A'<=L5<='Z') THEN
     L1_3 = L1]]L4]]L5;
 ELSE IF ('A'<=L2<='Z') & ('A'<=L3<='Z') & ('A'<=L4<='Z') THEN
     L1_3 = L2]]L3]]L4;
 ELSE IF ('A'<=L2<='Z') & ('A'<=L3<='Z') & ('A'<=L5<='Z') THEN
     L1_3 = L2]]L3]]L5;
 ELSE IF ('A'<=L3<='Z') & ('A'<=L4<='Z') & ('A'<=L5<='Z') THEN
     L1_3 = L3]]L4]]L5;
 ELSE L1_3 = L1]]L2]]L3;

 F1 = SUBSTR(FNAME,1,1);

 ID_KEY_3 = DOB]]L1_3]]F1;

 OUTPUT B;

END;

 /*****************************************************************/
 /* IF THERE IS NOT A FEASIBLE DATE OF BIRTH, ASSIGN FLAG VALUE */
 /*****************************************************************/

ELSE DO;
 MU_FL_3=2;
 OUTPUT C;
END;

RUN;

PROC SORT DATA=B; BY ID_KEY_3;
RUN;

DATA B1; SET B;
        BY ID_KEY_3;

 /*********************************/
 /* UNIQUE ID_KEY FLAG ASSIGNMENT */
 /*********************************/

IF FIRST.ID_KEY_3 & LAST.ID_KEY_3 THEN MU_FL_3=3;
```

```
 /***********************************/
 /* MULTIPLE ID_KEY FLAG ASSIGNMENT */
 /***********************************/

 ELSE MU_FL_3=4;
RUN;

PROC SORT DATA=B1; BY UPIN;
RUN;

 /*************************/
 /* PUT DATA BACK TOGETHER */
 /*************************/

DATA OUT.KEY3; SET B1 C;
          BY UPIN;
RUN;

PROC FREQ DATA=OUT.KEY3; TABLES MU_FL_3;
RUN;

PROC CONTENTS POSITION DATA=OUT.KEY3;
RUN;
```

APPENDIX D-4

ID_KEY_4 = DOB,L1_3

```
  /*********************************************************/
  /*                                                       */
  /* -CHECK FOR MULTIPLE UPINS PER PHYSICIAN FOR ACTIVE    */
  /*  PRACTICE SETTINGS.                                   */
  /*                                                       */
  /*********************************************************/

DATA A;
     SET IN.JUL0293
     (WHERE=(DOB>0)
      KEEP=UPIN INTERN SETDATE DOB LNAME);

  LNAME=LEFT(LNAME);

 /*****************************************/
 /* BOGUS OBSERVATION REMOVED BY CODE BELOW */
 /*****************************************/

 IF SETDATE>0 THEN SETYR = SUBSTR(SETDATE,7,2);

 IF SETYR='89' THEN SETYRADD=1;
 ELSE IF SETYR='90' ] SETYR='91' THEN SETYRADD=2;
 ELSE IF SETYR='92' ] SETYR='93' THEN SETYRADD=3;
 ELSE DELETE;

 /*****************************************************************/
 /* ACTIVE (OR MISSING) PRACTICE SETTINGS SORTED BY UPIN & SETDATE */
 /*****************************************************************/

IF (INTERN='R' ] INTERN='I' ] INTERN='P') THEN DRIP='A';
 ELSE IF INTERN='D' THEN DRIP='D';
 ELSE DRIP='M';

IF DRIP IN ('A','M');

RUN;

PROC SORT DATA=A; BY UPIN SETDATE;
RUN;

DATA B(KEEP=UPIN ID_KEY_4 MU_FL_4)
     C(KEEP=UPIN MU_FL_4);
       SET A(KEEP=UPIN DOB LNAME);
        BY UPIN;
        IF FIRST.UPIN;

INFORMAT MU_FL_4 1.;
```

```
/*************************/
/* CODE FOR FEASIBLE DOB */
/*************************/

 DOBYR = SUBSTR(DOB,7,2);
 DOBMO = SUBSTR(DOB,9,2);
 MODAY = SUBSTR(DOB,9,4);
 IF DOBYR='  ' THEN DO;
  IF (101<=DOB<=131) ] (201<=DOB<=229) ]
     (301<=DOB<=331) ] (401<=DOB<=430) ]
     (501<=DOB<=531) ] (601<=DOB<=630) ]
     (701<=DOB<=731) ] (801<=DOB<=831) ]
     (901<=DOB<=930) ] (1001<=DOB<=1031) ]
     (1101<=DOB<=1130) ] (1201<=DOB<=1231)
     THEN DOBYR='00';
  END;

IF DOBYR='00' THEN BYRH=0;
ELSE IF ' 1'<=DOBYR<'20' THEN BYRH=1;
ELSE IF '20'<=DOBYR<='69' THEN BYRH=2;
ELSE IF '70'<=DOBYR<='98' THEN BYRH=3;
ELSE IF DOBYR='99' & ('  '<DOBMO<'13') THEN BYRH=4;
ELSE IF DOBYR='99' THEN BYRH=5;
ELSE IF DOB=0 THEN BYRH=6;

IF BYRH=. ] BYRH=5 ] BYRH=6 THEN BYRHX=1;
ELSE IF BYRH=3 THEN BYRHX=2;
ELSE IF BYRH=4 THEN BYRHX=3;
ELSE BYRHX=4;

IF BYRHX=4 & MODAY^='0000' & DOB^=10101 &
    (('  101'<=MODAY<='0131') ] (' 201'<=MODAY<='0229') ]
     (' 301'<=MODAY<='0331') ] (' 401'<=MODAY<='0430') ]
     (' 501'<=MODAY<='0531') ] (' 601'<=MODAY<='0630') ]
     (' 701'<=MODAY<='0731') ] (' 801'<=MODAY<='0831') ]
     (' 901'<=MODAY<='0930') ] ('1001'<=MODAY<='1031') ]
     ('1101'<=MODAY<='1130') ] ('1201'<=MODAY<='1231')) THEN DO;

 /*************************************************************************/
 /* IF THERE IS A FEASIBLE DATE OF BIRTH, PASS THROUGH ID_KEY ALGORITHM */
 /*************************************************************************/

 INFORMAT L1-L5 $1. L1_3 $3.;

 L1 = SUBSTR(LNAME,1,1);
 L2 = SUBSTR(LNAME,2,1);
 L3 = SUBSTR(LNAME,3,1);
 L4 = SUBSTR(LNAME,4,1);
 L5 = SUBSTR(LNAME,5,1);
```

```
 IF ('A'<=L1<='Z') & ('A'<=L2<='Z') & ('A'<=L3<='Z') THEN
     L1_3 = L1]]L2]]L3;
 ELSE IF ('A'<=L1<='Z') & ('A'<=L2<='Z') & ('A'<=L4<='Z') THEN
     L1_3 = L1]]L2]]L4;
 ELSE IF ('A'<=L1<='Z') & ('A'<=L2<='Z') & ('A'<=L5<='Z') THEN
     L1_3 = L1]]L2]]L5;
 ELSE IF ('A'<=L1<='Z') & ('A'<=L3<='Z') & ('A'<=L4<='Z') THEN
     L1_3 = L1]]L3]]L4;
 ELSE IF ('A'<=L1<='Z') & ('A'<=L4<='Z') & ('A'<=L5<='Z') THEN
     L1_3 = L1]]L4]]L5;
 ELSE IF ('A'<=L2<='Z') & ('A'<=L3<='Z') & ('A'<=L4<='Z') THEN
     L1_3 = L2]]L3]]L4;
 ELSE IF ('A'<=L2<='Z') & ('A'<=L3<='Z') & ('A'<=L5<='Z') THEN
     L1_3 = L2]]L3]]L5;
 ELSE IF ('A'<=L3<='Z') & ('A'<=L4<='Z') & ('A'<=L5<='Z') THEN
     L1_3 = L3]]L4]]L5;
 ELSE L1_3 = L1]]L2]]L3;

 ID_KEY_4 = DOB]]L1_3;

 OUTPUT B;

END;

 /*****************************************************************/
 /* IF THERE IS NOT A FEASIBLE DATE OF BIRTH, ASSIGN FLAG VALUE */
 /*****************************************************************/

ELSE DO;
 MU_FL_4=2;
 OUTPUT C;
END;

RUN;

PROC SORT DATA=B; BY ID_KEY_4;
RUN;

DATA B1; SET B;
        BY ID_KEY_4;

 /*********************************/
 /* UNIQUE ID_KEY FLAG ASSIGNMENT */
 /*********************************/

IF FIRST.ID_KEY_4 & LAST.ID_KEY_4 THEN MU_FL_4=3;
```

```
 /***********************************/
 /* MULTIPLE ID_KEY FLAG ASSIGNMENT */
 /***********************************/

 ELSE MU_FL_4=4;
RUN;

PROC SORT DATA=B1; BY UPIN;
RUN;

 /*************************/
 /* PUT DATA BACK TOGETHER */
 /*************************/

DATA OUT.KEY4; SET B1 C;
          BY UPIN;
RUN;

PROC FREQ DATA=OUT.KEY4; TABLES MU_FL_4;
RUN;

PROC CONTENTS POSITION DATA=OUT.KEY4;
RUN;
```

APPENDIX D-5

ID_KEY_S = L1_3

```
  /*******************************************************/
  /*                                                     */
  /* -CHECK FOR MULTIPLE UPINS PER PHYSICIAN FOR ACTIVE  */
  /*  PRACTICE SETTINGS.                                 */
  /*                                                     */
  /*******************************************************/

DATA A;
     SET IN.JUL0293
     (WHERE=(DOB>0)
      KEEP=UPIN INTERN SETDATE DOB LNAME);

  LNAME=LEFT(LNAME);

 /*****************************************/
 /* BOGUS OBSERVATION REMOVED BY CODE BELOW */
 /*****************************************/

 IF SETDATE>0 THEN SETYR = SUBSTR(SETDATE,7,2);

 IF SETYR='89' THEN SETYRADD=1;
 ELSE IF SETYR='90' ] SETYR='91' THEN SETYRADD=2;
 ELSE IF SETYR='92' ] SETYR='93' THEN SETYRADD=3;
 ELSE DELETE;

 /*****************************************************************/
 /* ACTIVE (OR MISSING) PRACTICE SETTINGS SORTED BY UPIN & SETDATE */
 /*****************************************************************/

IF (INTERN='R' ] INTERN='I' ] INTERN='P') THEN DRIP='A';
 ELSE IF INTERN='D' THEN DRIP='D';
 ELSE DRIP='M';

IF DRIP IN ('A','M');

RUN;

PROC SORT DATA=A; BY UPIN SETDATE;
RUN;

DATA B(KEEP=UPIN ID_KEY_S);
       SET A(KEEP=UPIN DOB LNAME);
        BY UPIN;
        IF FIRST.UPIN;

 /************************/
 /* CODE FOR FEASIBLE DOB */
 /************************/
```

```
  DOBYR = SUBSTR(DOB,7,2);
  DOBMO = SUBSTR(DOB,9,2);
  MODAY = SUBSTR(DOB,9,4);
  IF DOBYR='  ' THEN DO;
   IF (101<=DOB<=131) ] (201<=DOB<=229) ]
      (301<=DOB<=331) ] (401<=DOB<=430) ]
      (501<=DOB<=531) ] (601<=DOB<=630) ]
      (701<=DOB<=731) ] (801<=DOB<=831) ]
      (901<=DOB<=930) ] (1001<=DOB<=1031) ]
      (1101<=DOB<=1130) ] (1201<=DOB<=1231)
      THEN DOBYR='00';
   END;

IF DOBYR='00' THEN BYRH=0;
ELSE IF ' 1'<=DOBYR<'20' THEN BYRH=1;
ELSE IF '20'<=DOBYR<='69' THEN BYRH=2;
ELSE IF '70'<=DOBYR<='98' THEN BYRH=3;
ELSE IF DOBYR='99' & ('  '<DOBMO<'13') THEN BYRH=4;
ELSE IF DOBYR='99' THEN BYRH=5;
ELSE IF DOB=0 THEN BYRH=6;

IF BYRH=. ] BYRH=5 ] BYRH=6 THEN BYRHX=1;
ELSE IF BYRH=3 THEN BYRHX=2;
ELSE IF BYRH=4 THEN BYRHX=3;
ELSE BYRHX=4;

IF BYRHX=4 & MODAY^='0000' & DOB^=10101 &
    ((' 101'<=MODAY<='0131') ] (' 201'<=MODAY<='0229') ]
     (' 301'<=MODAY<='0331') ] (' 401'<=MODAY<='0430') ]
     (' 501'<=MODAY<='0531') ] (' 601'<=MODAY<='0630') ]
     (' 701'<=MODAY<='0731') ] (' 801'<=MODAY<='0831') ]
     (' 901'<=MODAY<='0930') ] ('1001'<=MODAY<='1031') ]
     ('1101'<=MODAY<='1130') ] ('1201'<=MODAY<='1231')) THEN DELETE;

/*************************************************************************/
/* IF THERE IS NOT A FEASIBLE DATE OF BIRTH, PASS THROUGH ID_KEY ALGORITHM */
/*************************************************************************/

 INFORMAT L1-L5 $1. ID_KEY_S $3.;

 L1 = SUBSTR(LNAME,1,1);
 L2 = SUBSTR(LNAME,2,1);
 L3 = SUBSTR(LNAME,3,1);
 L4 = SUBSTR(LNAME,4,1);
 L5 = SUBSTR(LNAME,5,1);

 IF ('A'<=L1<='Z') & ('A'<=L2<='Z') & ('A'<=L3<='Z') THEN
     ID_KEY_S = L1]]L2]]L3;
 ELSE IF ('A'<=L1<='Z') & ('A'<=L2<='Z') & ('A'<=L4<='Z') THEN
     ID_KEY_S = L1]]L2]]L4;
```

```
 ELSE IF ('A'<=L1<='Z') & ('A'<=L2<='Z') & ('A'<=L5<='Z') THEN
     ID_KEY_S = L1]]L2]]L5;
 ELSE IF ('A'<=L1<='Z') & ('A'<=L3<='Z') & ('A'<=L4<='Z') THEN
     ID_KEY_S = L1]]L3]]L4;
 ELSE IF ('A'<=L1<='Z') & ('A'<=L4<='Z') & ('A'<=L5<='Z') THEN
     ID_KEY_S = L1]]L4]]L5;
 ELSE IF ('A'<=L2<='Z') & ('A'<=L3<='Z') & ('A'<=L4<='Z') THEN
     ID_KEY_S = L2]]L3]]L4;
 ELSE IF ('A'<=L2<='Z') & ('A'<=L3<='Z') & ('A'<=L5<='Z') THEN
     ID_KEY_S = L2]]L3]]L5;
 ELSE IF ('A'<=L3<='Z') & ('A'<=L4<='Z') & ('A'<=L5<='Z') THEN
     ID_KEY_S = L3]]L4]]L5;
 ELSE ID_KEY_S = L1]]L2]]L3;

RUN;

PROC SORT DATA=B; BY ID_KEY_S;
RUN;

DATA B1; SET B;
        BY ID_KEY_S;

INFORMAT MU_FL_S 1.;

 /*********************************/
 /* UNIQUE ID_KEY FLAG ASSIGNMENT */
 /*********************************/

IF FIRST.ID_KEY_S & LAST.ID_KEY_S THEN MU_FL_S=3;

 /***********************************/
 /* MULTIPLE ID_KEY FLAG ASSIGNMENT */
 /***********************************/

 ELSE MU_FL_S=4;
RUN;

PROC SORT DATA=B1; BY UPIN;
RUN;
```

```
 /*************************/
 /* PUT DATA BACK TOGETHER */
 /*************************/

DATA OUT.KEYS; SET B1;
          BY UPIN;
RUN;

PROC FREQ DATA=OUT.KEYS; TABLES MU_FL_S;
RUN;

PROC CONTENTS POSITION DATA=OUT.KEYS;
RUN;
```

APPENDIX E

MERGING PROGRAMS

APPENDIX E-1

PROGRAM TO MERGE MULTIPLE UPIN INDICATORS WITH MAIN ANALYTIC FILE

```
  /*************************************************************/
  /*                                                           */
  /* DATA INE DSN=TI04.@DCH7891.UPIN.JUL0293.EDIT01(EDIT01)    */
  /*      IN1 DSN=TI04.@DCH7891.UPIN.JUL0293.KEY1(KEY1)        */
  /*      IN2 DSN=TI04.@DCH7891.UPIN.JUL0293.KEY2(KEY2)        */
  /*      IN3 DSN=TI04.@DCH7891.UPIN.JUL0293.KEY3(KEY3)        */
  /*      IN4 DSN=TI04.@DCH7891.UPIN.JUL0293.KEY4(KEY4)        */
  /*      INS DSN=TI04.@DCH7891.UPIN.JUL0293.KEYS(KEYS)        */
  /*      OUT DSN=TI04.@DCH7891.UPIN.JUL0293.EDIT02(EDIT02)    */
  /*                                                           */
  /* -MERGING TOGETHER THE FINAL RESEARCHER EDITS DATABASE     */
  /*                                                           */
  /*************************************************************/

DATA OUT.EDIT02;
     MERGE INE.EDIT01
           IN1.KEY1
           IN2.KEY2
           IN3.KEY3
           IN4.KEY4
           INS.KEYS;
     BY UPIN;

 /************************************************/
 /* ASSIGNMENT OF NO ACTIVE PRACTICE SETTING FLAG */
 /************************************************/

 ARRAY FL MU_FL_1-MU_FL_4;

       DO OVER FL;
          IF FL=. THEN FL=1;
       END;

 IF MU_FL_1=1 THEN MU_FL_S=1;

 /*********************************************************************/
 /* ASSIGNMENT OF INFEASIBLE DOB FLAG TO ZERO FOR THE CASES THAT HAD */
 /* A FEASIBLE DOB                                                    */
 /*********************************************************************/

 IF MU_FL_1 IN (3,4) THEN MU_FL_S=0;
RUN;

PROC FREQ; TABLES MU_FL_1-MU_FL_4 MU_FL_S;
RUN;

PROC CONTENTS POSITION;
RUN;
```

APPENDIX E-2

PROGRAM TO MERGE EDITED FILE WITH ORIGINAL FILE

```
  /*************************************************************/
  /*                                                           */
  /* DATA IN1 DSN=TIO4.@DCH7891.UPIN.JUL0293(JUL0293)          */
  /*      IN2 DSN=TIO4.@DCH7891.UPIN.JUL0293.EDIT02(EDIT02)    */
  /*      OUT DSN=TIO4.@DCH7891.UPIN.JUL0293.WITHEDIT(WITHEDIT)*/
  /*                                                           */
  /* -MERGING THE ORIGINAL DATA WITH THE EDITED DATABASE       */
  /*                                                           */
  /*************************************************************/

PROC SORT DATA=IN1.JUL0293
 OUT=J(RENAME=(MEDICARE=CPIN));
 BY OBSNUM;

PROC SORT DATA=IN2.EDIT02
 OUT=E;
 BY OBSNUM;

DATA OUT.WITHEDIT;
     MERGE J E;
     BY OBSNUM;
RUN;
PROC CONTENTS POSITION;
RUN;
```

APPENDIX F

MEDICAL SCHOOL ESTABLISHMENT AND CLOSURE DATES

APPENDIX F

MEDICAL SCHOOL ESTABLISHMENT AND CLOSURE DATES

HCFA ID	MEDICAL SCHOOL	LOCATION	YEAR OPEN	YEAR CLOSE
00102	University of Alabama School of Medicine	Birmingham, AL	1859	3000
00104	Birmingham Medical College	Birmingham, AL	1894	1915
00106	University of South Alabama College of Medicine	Mobile, AL	1967	3000
00301	University of Arizona College of Medicine	Tuscon, AR	1962	3000
00401	University of Arkansas College of Medicine	Little Rock, AR	1879	3000
00402	College of Physicians and Surgeons	Little Rock, AR	1906	1911
00501	Cooper Medical College	San Francisco, CA	1882	1912
00502	University of California, San Francisco, School of Medicine	San Francisco, CA	1864	3000
00504	California Eclectic Medical College	Los Angeles, CA	1907	1915
00505	Hahneman Medical College of the Pacific (Homeopathic)	San Francisco, CA	1902	1918
00506	University of Southern California School of Medicine	Los Angeles, CA	1885	3000
00507	College of Physicians and Surgeons of San Francisco	San Francisco, CA	1896	1921
00508	Oakland College of Medicine and Surgery	Oakland, CA	1900	1920
00509	College of Physicians and Surgeons of Los Angeles	Los Angeles, CA	1904	1909
00511	Stanford University School of Medicine	Palo Alto, CA	1908	3000
00512	Loma Linda University School of Medicine	Loma Linda, CA	1909	3000
00513	Pacific Medical College	Los Angeles, CA	1913	1916
00514	University of California, Los Angeles, School of Medicine	Los Angeles, CA	1951	3000
00515	University of California, Irvine, California College of Medicine	Irvine, CA	1962	3000
00516	University California Ca College of Med Irvine CA	Irvine, CA	1962	3000
00517	University Of Cal College of Medicine Irvine CA	Irvine, CA	1962	3000
00518	University of California, San Diego, School of Medicine	La Jolla, CA	1962	3000
00519	University of California, Davis, School of Medicine	Davis, CA	1963	3000
00702	University of Colorado School of Medicine	Denver, CO	1883	3000
00705	Denver and Gross College of Medicine	Denver, CO	1902	1911
00801	Yale University School of Medicine	New Haven, CT	1810	3000
00802	University of Connecticut School of Medicine	Farmington, CT	1961	3000
01001	George Washington University School of Medicine and Health Sciences	Washington, D.C.	1825	3000
01002	Georgetown University School of Medicine	Washington, D.C.	1851	3000
01003	Howard University College of Medicine	Washington, D.C.	1868	3000
01102	University of Miami School of Medicine	Miami, FL	1952	3000
01103	University of Florida College of Medicine	Gainesville, FL	1956	3000
01104	University of South Florida College of Medicine	Tampa, FL	1965	3000
01201	Medical College of Georgia	Augusta, GA	1828	3000
01205	Emory University School of Medicine	Augusta, GA	1854	3000
01209	Georgia College of Eclectic Medicine and Surgery	Augusta, GA	1886	1916
01211	Atlanta College of Physicians and Surgeons	Atlanta, GA	1898	1913
01212	Atlanta School of Medicine	Atlanta, GA	1905	1913
01218	Hospital Medical College Eclectic	Atlanta, GA	1908	1911
01219	Southern College of Medicine & Surgery	Atlanta, GA	1911	1914
01221	Morehouse School of Medicine	Atlanta, GA	1978	3000
01222	Mercer University School of Medicine	Macon, GA	1973	3000
01401	University of Hawaii John A. Burns School of Medicine	Honolulu, HI	1967	3000
01601	Rush Medical College	Chicago, IL	1837	3000
01602	University of Chicago Pritzker School of Medicine	Chicago, IL	1924	3000
01604	Hahneman Medical College and Hospital	Chicago, IL	1855	1922
01605	College Of Medicine and Surgery (Physio-Medical)	Chicago, IL	1899	1911

APPENDIX F (continued)

MEDICAL SCHOOL ESTABLISHMENT AND CLOSURE DATES

HCFA ID	MEDICAL SCHOOL	LOCATION	YEAR OPEN	YEAR CLOSE
01606	Northwestern University Medical School	Chicago, IL	1859	3000
01608	Bennett Medical College	Chicago, IL	1909	1915
01609	Northwestern University Womans Medical School	Chicago, IL	1892	1902
01610	The Chicago Homeopathic Medical College	Chicago, IL	1877	1904
01611	University of Illinois College of Medicine	Chicago, IL	1881	3000
01613	Harvey Medical College	Chicago, IL	1891	1905
01614	National Medical University	Chicago, IL	1891	1914
01615	Hering Medical College (Homeopathic)	Chicago, IL	1892	1913
01616	Jenner Medical College	Chicago, IL	1896	1917
01617	Illinois Medical College	Chicago, IL	1894	1910
01618	Dunham Medical College	Chicago, IL	1896	1902
01619	American Medical Missionary College	Battle Creek, MI & Chicago, IL	1895	1910
01622	Chicago College of Medicine and Surgery	Chicago, IL	1907	1917
01623	Dearborn Medical College	Chicago, IL	1903	1907
01642	University of Health Sciences/The Chicago Medical School	North Chicago, IL	1912	3000
01643	Loyola University of Chicago, Stritch School of Medicine	Maywood, IL	1915	3000
01644	The General Medical College	Chicago, IL	1922	1924
01645	Southern Illinois University School of Medicine	Springfield, IL	1969	3000
01705	Physio-Medical College of Indiana	Indianapolis, IN	1873	1909
01708	The Medical College of Indiana	Indianapolis, IN	1878	1905
01709	Central College of Physicians & Surgeons	Indianapolis, IN	1879	1905
01717	Eclectic Medical College of Indiana	Indianapolis, IN	1900	1908
01718	Indiana Medical College, School of Medicine of Purdue University	Indianapolis, IN	1905	1908
01720	Indiana University School of Medicine	Indianapolis, IN	1903	3000
01801	College of Physicians and Surgeons	Keokuk, IA	1849	1899
01803	University of Iowa College of Medicine	Iowa City, IA	1850	3000
01804	State University of Iowa, College of Homeopathic Medicine	Iowa City, IA	1877	1919
01806	Drake University College of Medicine	Des Moines, IA	1903	1913
01808	Sioux City College of Medicine	Sioux City, IA	1891	1909
01810	Keokuk Medical College, College of Physicians and Surgeons	Keokuk, IA	1899	1908
01902	University of Kansas School of Medicine	Kansas City, KS	1899	3000
01903	Kansas Medical College (Medical Department of Washburn College)	Topeka, KS	1890	1913
01904	College of Physicians and Surgeons of Kansas City	Kansas City, KS	1894	1905
01907	Western Eclectic College of Medicine and Surgery	Kansas City, KS	1908	1909
02001	Kentucky School of Medicine	Louisville, KY	1850	1908
02002	University Of Louisville School of Medicine	Louisville, KY	1833	3000
02004	Louisville Medical College	Louisville, KY	1869	1907
02005	Hospital College of Medicine (Med Dept of Central University)	Louisville, KY	1873	1907
02007	Louisville National Medical College (Med Dept State University)(Black)	Louisville, KY	1888	1912
02008	Southwestern Homeopathic Medical College and Hospital	Louisville, KY	1893	1910
02009	Kentucky University Medical Department	Louisville, KY	1898	1907
02011	Louisville and Hospital Medical College	Louisville, KY	1907	1908
02012	University of Kentucky College of Medicine	Lexington, KY	1954	3000
02101	Tulane University School of Medicine	New Orleans, LA	1834	3000
02104	Flint Medical College of New Orleans University (Black)	New Orleans, LA	1901	1911
02105	Louisiana State University School of Medicine in New Orleans	New Orleans, LA	1931	3000
02106	Louisiana State University School of Medicine in Shreveport	Shreveport, LA	1966	3000

APPENDIX F (continued)

MEDICAL SCHOOL ESTABLISHMENT AND CLOSURE DATES

HCFA ID	MEDICAL SCHOOL	LOCATION	YEAR OPEN	YEAR CLOSE
02201	Bowdoin Medical School	Brunswick and Portland, ME	1915	1921
02301	University of Maryland School of Medicine	Baltimore, MD	1807	3000
02303	College of Physicians and Surgeons of Baltimore	Baltimore, MD	1872	1915
02304	Baltimore Medical College	Baltimore, MD	1881	1913
02305	Woman's Medical College of Baltimore	Baltimore, MD	1882	1910
02306	Baltimore University School of Medicine	Baltimore, MD	1884	1907
02307	John Hopkins University School of Medicine	Baltimore, MD	1893	3000
02308	Atlantic Medical College	Baltimore, MD	1907	1910
02309	Maryland Medical College	Baltimore, MD	1898	1913
02311	Maryland College of Eclectic Medicine and Surgery	Baltimore, MD	1914	1915
02312	Uniformed Services University of Health Science	Bethesda, MD	1972	3000
02401	Harvard Medical School	Boston, MA	1782	3000
02405	Boston University School of Medicine	Boston, MA	1873	3000
02406	College of Physicians and Surgeons	Boston, MA	1880	1949
02407	Tufts University School of Medicine	Boston, MA	1893	3000
02415	Middlesex University School of Medicine	Waltham, MA	1937	1946
02416	University of Massachusetts Medical School	Worcester, MA	1962	3000
02501	University of Michigan Medical School	Ann Arbor, MI	1850	3000
02505	University of Michigan Homeopathic Medical School	Ann Arbor, MI	1875	1922
02507	Wayne State University School of Medicine	Detroit, MI	1868	3000
02508	Michigan College of Medicine & Surgery	Detroit, MI	1888	1907
02512	Michigan State University College of Human Medicine	East Lansing, MI	1964	3000
02604	University of Minnesota Medical School	Minneapolis, MN	1888	3000
02605	Minneapolis College of Physicians and Surgeons	Minneapolis, MN	1883	1908
02607	University of Minnesota Duluth School of Medicine	Duluth, MN	1969	3000
02608	Mayo Medical School	Rochester, MN	1971	3000
02701	University Mississippi School of Medicine	Jackson, MI	1903	3000
02702	Mississippi Medical College	Meridian, MI	1906	1912
02801	Missouri Medical College	St. Louis, MO	1840	1899
02802	Washington University Medical School	St. Louis, MO	1855	3000
02803	University of Missouri Columbia School of Medicine	Columbia, MO	1872	3000
02805	Homeopathic Medical College of Missouri	St. Louis, MO	1859	1909
02807	St. Louis College of Physicians and Surgeons	St. Louis, MO	1879	1917
02808	Kansas City Medical College	Kansas City, MO	1880	1905
02810	National University of the Arts and Sciences Medical Department	St. Louis, MO	1912	1918
02820	University Medical College of Kansas City	Kansas City, MO	1888	1913
02822	Ensworth Medical College	St. Joseph, MO	1888	1914
02826	Kansas City Homeopathic Medical College	Kansas City, MO	1888	1902
02828	Barnes Medical College	St. Louis, MO	1892	1911
02830	Woman's Medical College	Kansas City, MO	1895	1903
02833	Eclectic Medical University	Kansas City, MO	1898	1918
02834	Saint Louis University School of Medicine	St. Louis, MO	1901	3000
02835	Southwest School of Medicine and Hospital	Kansas City, MO	1915	1916
02843	Kansas City College of Medicine and Surgery	Kansas City, MO	1915	1926
02844	Kansas City University of Physicians and Surgeons	Kansas City, MO	1918	1944
02845	Mid West Medical College	Kansas City, MO	1933	1940
02846	University of Missouri Kansas City School of Medicine	Kansas City, MO	1969	3000

APPENDIX F (continued)

MEDICAL SCHOOL ESTABLISHMENT AND CLOSURE DATES

HCFA ID	MEDICAL SCHOOL	LOCATION	YEAR OPEN	YEAR CLOSE
03004	Lincoln Medical College, Eclectic	Lincoln, NE	1915	1918
03005	University of Nebraska College of Medicine	Omaha, NE	1880	3000
03006	Creighton University School of Medicine	Omaha, NE	1892	3000
03007	Nebraska College of Medicine	Lincoln, NE	1905	1909
03101	University of Nevada School of Medicine	Reno, NV	1969	3000
03201	Dartmouth Medical School	Hanover, NH	1797	3000
03305	University of Medicine and Dentistry of New Jersey-New Jersey Medical School	Newark, NJ	1956	3000
03306	University of Medicine and Dentistry of New Jersey- Robert Wood Johnson Medical School	Piscataway, NJ	1966	3000
03401	University of New Mexico School of Medicine	Albuquerque, NM	1961	3000
03501	Columbia University College of Physicians and Surgeons	New York, NY	1767	3000
03503	Albany Medical College of Union University	Albany, NY	1839	3000
03506	State University of New York at Buffalo School of Medicine	Buffalo, NY	1846	3000
03508	State University of New York Health Science Center at Brooklyn	Brooklyn, NY	1860	3000
03509	New York Medical College	Valhalla, NY	1858	3000
03510	Bellevue Hospital Medical College	New York, NY	1861	1898
03511	New York Medical College and Hospital for Women, Homeopathic	New York, NY	1866	1918
03513	Eclectic Medical College of the City of New York	New York, NY	1866	1913
03515	State University of New York Health Science Center	Syracuse, NY	1834	3000
03519	New York University School of Medicine	New York, NY	1841	3000
03520	Cornell University Medical College	New York, NY	1898	3000
03543	Fordham University School of Medicine	New York, NY	1907	1921
03545	University of Rochester School of Medicine and Dentistry	Rochester, NY	1920	3000
03546	Albert Einstein College of Medicine of Yeshiva University	New York, NY	1955	3000
03547	Mount Sinai School of Medicine of the City University of New York	New York, NY	1963	3000
03548	State University of New York at Stony Brook School of Medicine	Stony Brook, NY	1962	3000
03601	University of North Carolina at Chapel Hill School of Medicine	Chapel Hill, NC	1879	3000
03603	Leonard Medical School (Black)	Raleigh, NC	1882	1918
03604	North Carolina Medical College	Charlotte, NC	1907	1918
03605	Bowman Gray School of Medicine of Wake Forest University	Winston-Salem, NC	1902	3000
03607	Duke University School of Medicine	Durham, NC	1925	3000
03608	East Carolina University School of Medicine	Greenville, NC	1967	3000
03701	University of North Dakota School of Medicine	Grand Forks, ND	1905	3000
03801	Medical College of Ohio	Cincinnati, OH	1819	1909
03802	Eclectic Medical College	Cincinnati, OH	1910	1942
03803	Starling Medical College	Columbus, OH	1847	1907
03806	Case Western Reserve University School of Medicine	Cleveland, OH	1843	3000
03808	Cincinnati College of Medicine and Surgery	Cincinnati, OH	1849	1902
03809	Miami Medical College	Cincinnati, OH	1852	1909
03811	University of Wooster Medical Department	Cleveland, OH	1869	1881
03819	Toledo Medical College (Med Dept of Toledo University)	Toledo, OH	1883	1914
03823	Cleveland Medical College, Homeopathic	Cleveland, OH	1890	1898
03825	Ohio Medical University	Columbus, OH	1890	1907
03826	Cleveland-Pulte Medical College	Cleveland, OH	1910	1914
03840	Ohio State University College of Medicine	Columbus, OH	1907	3000
03841	University of Cincinnati College of Medicine	Cincinnati, OH	1819	3000
03843	Medical College of Ohio at Toledo	Toledo, OH	1964	3000
03844	Northeastern Ohio Universities College of Medicine	Rootstown, OH	1973	3000

APPENDIX F (continued)

MEDICAL SCHOOL ESTABLISHMENT AND CLOSURE DATES

HCFA ID	MEDICAL SCHOOL	LOCATION	YEAR OPEN	YEAR CLOSE
03845	Wright State University School of Medicine	Dayton, OH	1973	3000
03901	University of Oklahoma College of Medicine	Oklahoma City, OK	1900	3000
03905	Oral Roberts University School of Medicine	Tulsa, OK	1979	1990
04001	Willamette University Medical Department	Salem, OR	1866	1913
04002	Oregon Health Sciences University School of Medicine	Portland, OR	1887	3000
04101	University of Pennsylvania School of Medicine	Philadelphia, PA	1765	3000
04102	Jefferson Medical College of Thomas Jefferson University	Philadelphia, PA	1824	3000
04107	Medical College of Pennsylvania	Philadelphia, PA	1850	3000
04109	Hahnemann University School of Medicine	Philadelphia, PA	1848	3000
04111	Medico-Chirurgical College of Philadelphia	Philadelphia, PA	1881	1916
04112	University of Pittsburgh School of Medicine	Pittsburgh, PA	1886	3000
04113	Temple University School of Medicine	Philadelphia, PA	1901	3000
04114	Pennsylvania State University College of Medicine	Hershey, PA	1963	3000
04201	University of Puerto Rico School of Medicine	San Juan, Puerto Rico	1949	3000
04202	Ponce School of Medicine	Ponce, Puerto Rico	1978	3000
04203	University Central del Caribe Escuela de Medicina	Cayey, Puerto Rico	1976	3000
04204	Ciencias Medicina de San Juan Bautista	Hato Rey, Puerto Rico	1978	3000
04301	Brown University Program in Medicine	Providence, RI	1963	3000
04501	Medical University of South Carolina College of Medicine	Charleston, SC	1824	3000
04504	University of South Carolina School of Medicine	Columbia, SC	1974	3000
04601	University of South Dakota School of Medicine	Vermillion, SD	1907	3000
04701	University of Nashville Medical Department	Nashville, TN	1850	1911
04705	Vanderbilt University School of Medicine	Nashville, TN	1874	3000
04706	U...versity of Tennessee College of Medicine	Memphis, TN	1851	3000
04707	Meharry Medical College School of Medicine	Nashville, TN	1876	3000
04708	Memphis Hospital Medical College	Memphis, TN	1878	1913
04709	Chattanooga Medical College	Chattanooga, TN	1889	1910
04710	Lincoln Memorial University Medical Department	Knoxville, TN	1909	1914
04711	Unversity of the South Medical Department	Sewanee, TN	1892	1909
04713	Knoxville Medical College (Black)	Knoxville, TN	1900	1910
04714	University of West Tennesee College of Medicine and Surgery (Black)	Memphis, TN	1907	1923
04715	College of Physicians and Surgeons	Memphis, TN	1906	1911
04720	East Tennessee State University James H. Dishner College of Medicine	Johnson City, TN	1978	3000
04802	University of Texas Medical Branch at Galveston	Galveston, TX	1881	3000
04803	Fort Worth School Of Medicine	Fort Worth, TX	1894	1918
04804	Baylor University College Of Medicine	Houston, TX	1900	3000
04805	Physio-Medical College of Texas	Dallas, TX	1902	1908
04806	Southern Methodist University Medical Department	Dallas, TX	1911	1915
04807	Gate City Medical College	Dallas, TX	1910	1911
04812	University of Texas Southwestern Medical Center at Dallas School of Medicine	Dallas, TX	1943	3000
04813	University of Texas Medical School at San Antonio	San Antonio, TX	1966	3000
04814	University of Texas Medical School at Houston	Houston, TX	1969	3000
04815	Texas Tech University Health Sciences Center School of Medicine	Lubbock, TX	1969	3000
04816	Texas A&M University College of Medicine	College Station, TX	1971	3000
04901	University of Utah School of Medicine	Salt Lake City, UT	1905	3000
05002	University of Vermont College of Medicine	Burlington, VT	1822	3000
05101	University of Virginia School of Medicine	Charlottesville, VA	1825	3000

APPENDIX F (continued)

MEDICAL SCHOOL ESTABLISHMENT AND CLOSURE DATES

HCFA ID	MEDICAL SCHOOL	LOCATION	YEAR OPEN	YEAR CLOSE
05104	Virginia Commonwealth University Medical College of Virginia	Richmond, VA	1838	3000
05106	University College of Medicine	Richmond, VA	1893	1913
05107	Eastern Virginia Medical School	Norfolk, VA	1973	3000
05404	University of Washington School of Medicine	Seattle, WA	1945	3000
05415	Washington College of Physicians and Surgeons	Seattle, WA		
05501	West Virginia University School of Medicine	Morgantown, WV	1902	3000
05502	Marshall University School of Medicine	Huntington, WV	1974	3000
05602	Wisconsin College of Physicians and Surgeons	Milwaukee, WI	1893	1912
05603	Milwaukee Medical College	Milwaukee, WI	1894	1912
05605	University of Wisconsin Medical School	Madison, WI	1907	3000
05606	Medical College of Wisconsin	Milwaukee, WI	1912	3000

Notes.

1. It appears that three codes have been assigned to the medical school at the University of California at Irvine (see shaded lines on first page of this table).
2. A value of 3000 in the year closed column indicates that the school is still open.
3. Open and close dates could not be found for the Washington College of Physicians and Surgeons (05415).

CMS LIBRARY
3 8095 00014113 1